MODULE – 5

Introduction

Local Replication

Replication is the process of creating an exact copy of data. Creating one or more replicas of the production data is one of the ways to provide Business Continuity (BC). These replicas can be used for recovery and restart operations in the event of data loss.

The primary purpose of replication is to enable users to have designated data at the right place, in a state appropriate to the recovery need. The replica should provide recoverability and restartability.

Recoverability enables restoration of data from the replicas to the production volumes in the event of data loss or data corruption. It must provide minimal RPO and RTO for resuming business operations on the production volumes, while restartability must ensure consistency of data on the replica. This enables restarting business operations using the replicas.

Replication can be classified into two major categories: local and remote. Local replication refers to replicating data within the same array or the same data center. Remote replication refers to replicating data at a remote site.

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 pro- duction 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.”

Uses of Local Replicas

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

1. **Alternate source for backup:** Under normal backup operations, data is read from the production volumes (LUNs) and written to the backup device. This places additional burden on the production infrastructure, as production LUNs are simultaneously involved in production work. As the local replica contains an exact point-in-time (PIT) copy of the source data, it can be used 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.
2. **Fast recovery:** In the event of a partial failure of the source, or data corruption, a local replica can be used to recover lost data. In the event of a complete failure of the source, the replica can be restored to a different set of source devices. In either case, this method provides faster recovery and minimal RTO, compared to traditional restores from tape backups. In many instances business operations can be started using the source device before the data is completely copied from thereplica.
3. **Decision-support activities such as reporting:** Running the reports using the data on the replicas greatly reduces the I/O burden placed on the productiondevice.
4. **Testing platform:** A local replica can be used for testing critical business data or applications. For example, when planning an application upgrade, it can be tested using the local replica. If the test is successful, it can be restored to the sourcevolumes.
5. **Data migration:** Local replication can also be used for data migration. Data migration may be performed for various reasons, such as migrating from a small LUN to a largerLUN.

Replica Consistency

Most file systems and databases buffer data in the host before it is written to disk. A consistent replica ensures that data buffered in the host is properly captured on the disk when the replica is created. Ensuring consistency is the primary requirement for all the replicationtechnologies.

Consistency of a Replicated File System

- File systems buffer data in host memory to improve application response time. The buffered information is periodically written to disk. In UNIX operating systems, the *sync daemon* is the process that flushes the buffers to disk at set intervals.
- In some cases, the replica may be created in between the set intervals. Hence, the host memory buffers must be flushed to ensure data consistency on the replica, prior to its creation.
- Figure 13-1 illustrates flushing of the buffer to its source, which is then replicated. If the host memory buffers are not flushed, data on the replica will not contain the information that was buffered in the host. If the file system is unmounted prior to the creation of the replica, the buffers would be automatically flushed and data would be consistent on the replica.

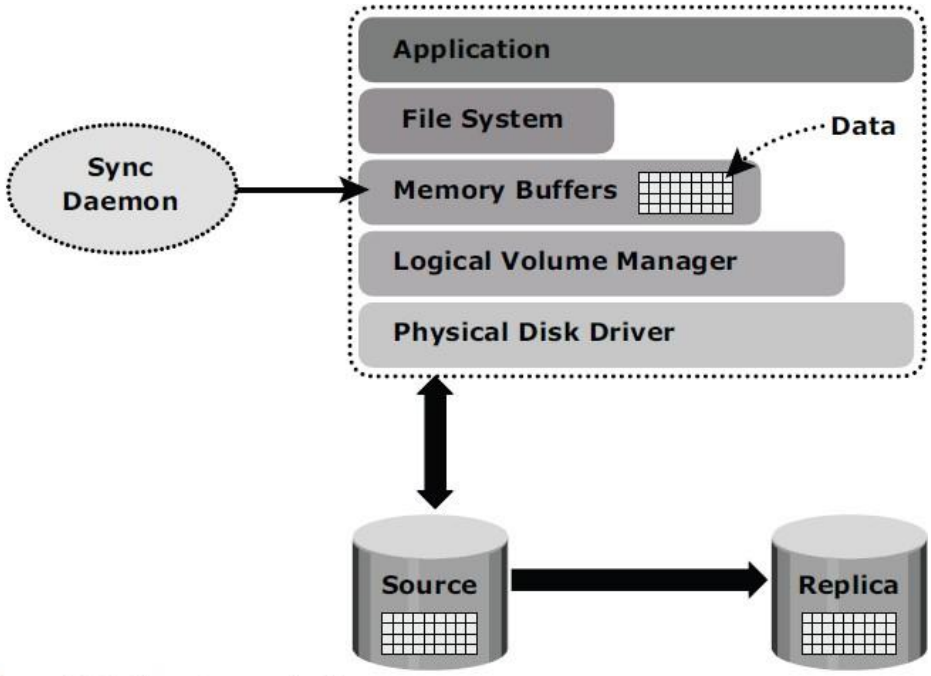


Figure 13-1: File system replication

- If a mounted file system is replicated, some level of recovery such as *fsck* or *log replay* would be required on the replicated file system. When the file system replication process is 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 can be performed with the database offline or online. If the database is offline, it is not available for I/O operations. Because no updates are occurring, the replica will be consistent.

If the database is online, it is available for I/O operations. Transactions to the database will be updating data continuously. When a database is backed up while it is online, changes made to the database at this time must be applied to the backup copy to make it consistent. Performing an online backup requires additional procedures during backup and restore. Often these procedures can be scripted to automate the process, alleviating administrative work and minimizing human error. Most databases support some form of online or hot backups. There will be increased logging activity during the time when the database is in the hot backupmode.

An alternate approach exploits the *dependent write I/O* principle inherent in any database management system (DBMS). 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. Dependent write consistency is required for protection against power outages, loss of local channel connectivity, or storage devices. When the failure occurs a dependent write consistent image is created. A restart transforms the dependent write consistent image to a transactional consistent image — i.e., committed transactions are recovered, and in-flight transactions are discarded.

In order for a transaction to be deemed complete, databases require that a series of writes have to occur in a particular order. These writes would be recorded on the various devices/file systems. Figure 13-2, illustrates the process of flushing the buffer from host to source; I/Os 1 to 4 must complete, in order for the transaction to be considered complete. I/O 4 is dependent on I/O 3 and will occur 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).

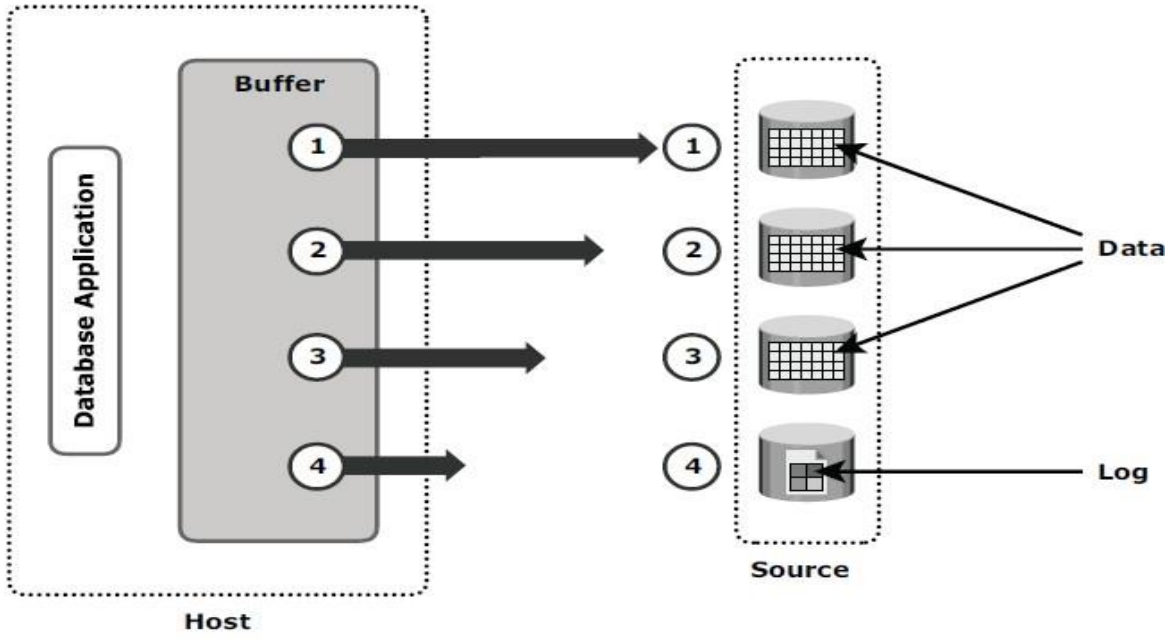


Figure 13-2: Dependent write consistency on sources

At the point in time when the replica is created, all the writes to the source devices must be captured on the replica devices to ensure data consistency. Figure 13-3 illustrates the process of replication from source to replica, I/O transactions 1 to 4 must be carried out in order for the data to be consistent on the replica.

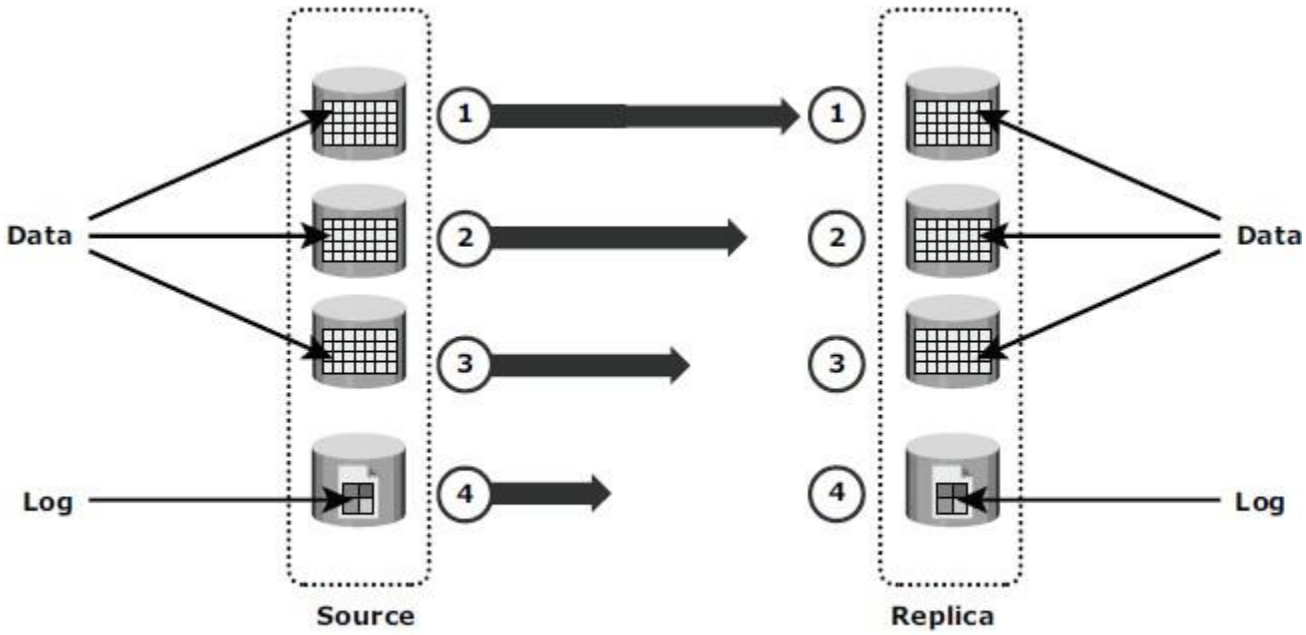


Figure 13-3: Dependent write consistency on replica

Creating a PIT copy for multiple devices happens quickly, but not instantaneously. 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. 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.

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.

Host-Based Local Replication

In host-based replication, logical volume managers (LVMs) or the file systems perform the local replication process. LVM-based replication and file system (FS) snapshot are examples of host-based local replication.

LVM-Based Replication

In LVM-based replication, logical volume manager is responsible for creating and controlling the host-level logical volume. An LVM has three components: physical volumes (physical disk), volume groups, and logical volumes. A *volume group* is created by grouping together 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 partition* in a logical volume is mapped to two physical partitions on two different physical volumes, as shown in Figure 13-4. An application write to a logical partition is written to the two physical partitions 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. LVM mirrors can be

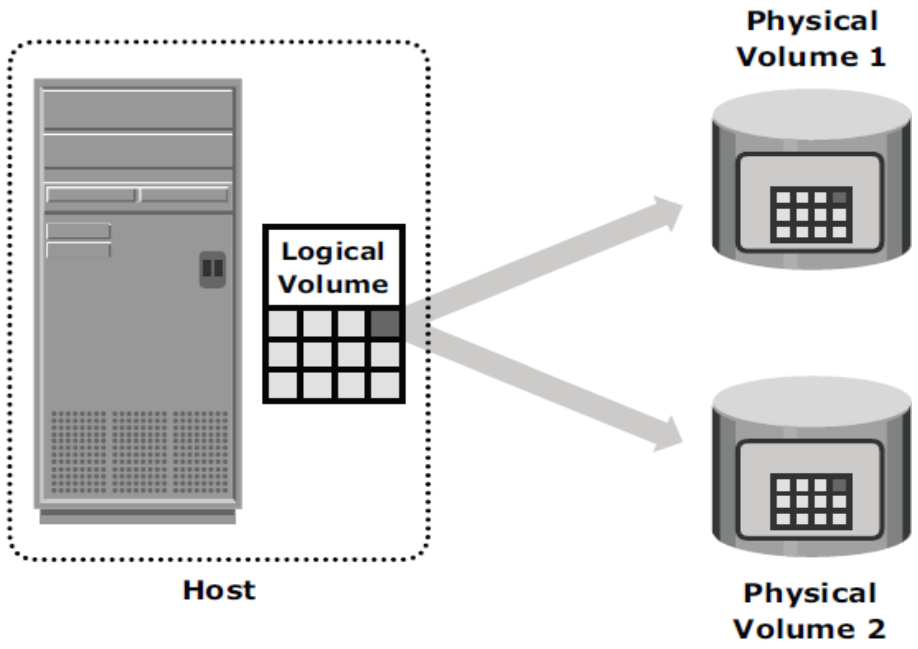


Figure 13-4: LVM-based mirroring

added or removed dynamically.

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.

Limitations of LVM-Based Replication

As every write generated by an application translates into two writes on the disk, an additional burden is placed on the host CPU. This can degrade application performance. Presenting an LVM-based local replica to a second 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.

File System Snapshot

File system (FS) snapshot is a pointer-based replica that requires a fraction of the space used by the original FS. This snapshot can be implemented by either FS itself or by LVM. It uses Copy on First Write (CoFW) principle.

When the snapshot is created, a bitmap and a 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 creation of the snap. The blockmap is used to indicate the exact address from which data is to be read when the data is accessed from the Snap FS. Immediately after creation of the snapshot all reads from the snapshot will actually be served by reading the production 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 produc- tion 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 pro- duction 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.

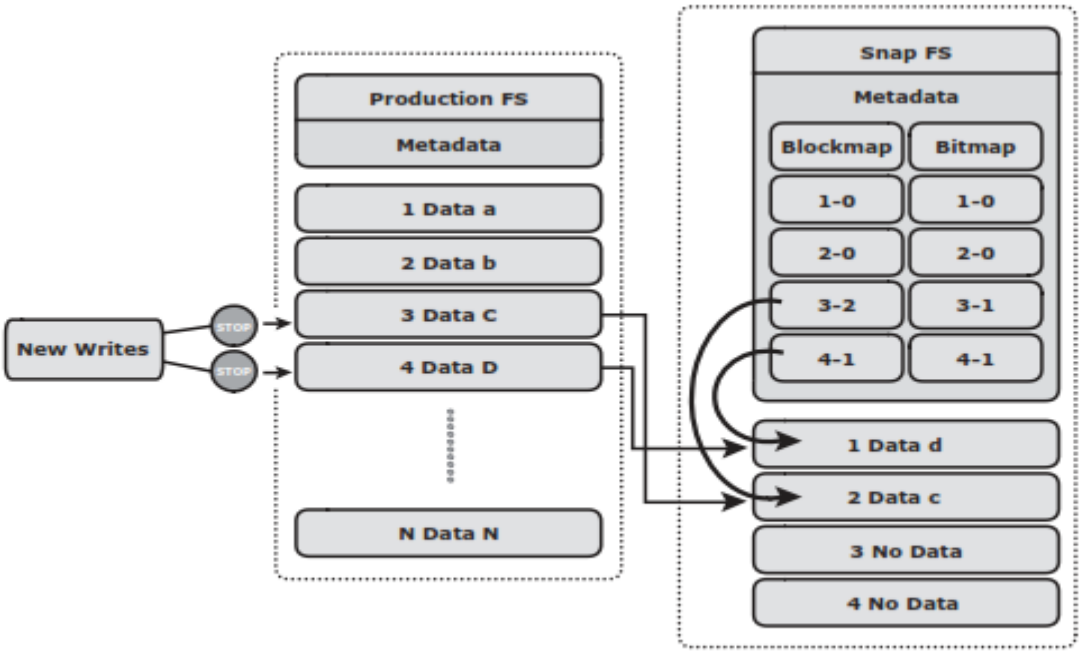


Figure 11-6: Write to production FS

Storage Array–Based Replication

In *storage array-based local replication*, the array operating environment performs the local replication process. The host resources such as 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 alternate host for any business operations.

In this replication, the required number of replica devices should be selected on the same array and then data is replicated between source-replica pairs. A database could be laid out over multiple physical volumes and in that case all the devices must be replicated for a consistent PIT copy of the database. Figure 13-5 shows storage array based local replication, where source and target are in the same array and accessed by different hosts.

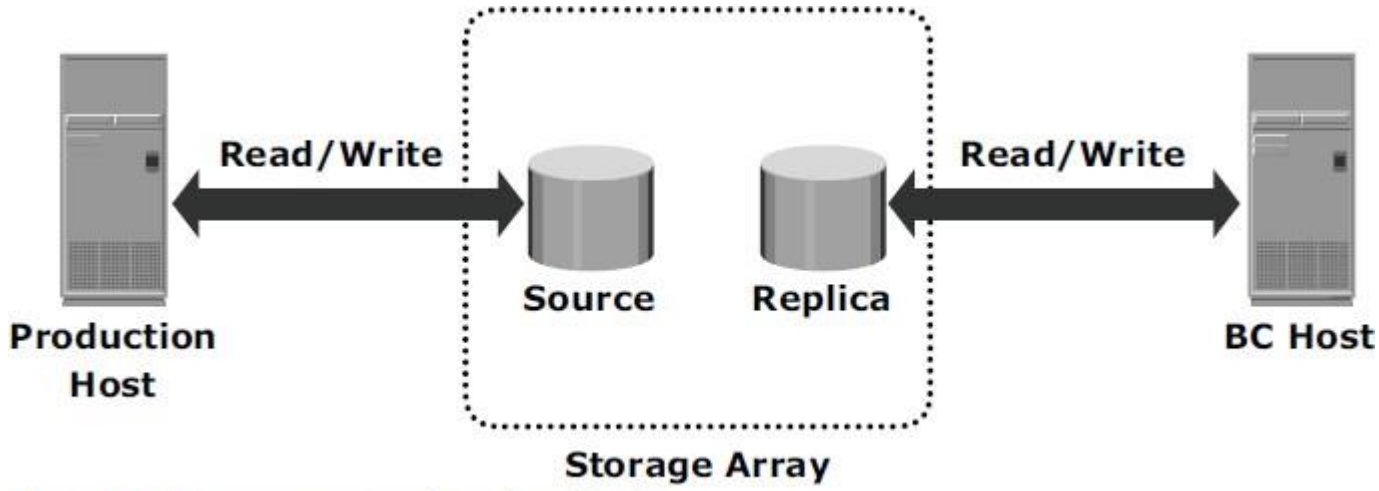


Figure 13-5: Storage array-based replication

Full-Volume Mirroring

In *full-volume mirroring*, the target is attached to the source and established as a mirror of the source shown in Figure 13-6 [a].
Existing data on the source is copied to the target. New updates to the source are also updated on thetarget.

After all the data is copied and both the source and the target contain identical data, the target can be considered a mirror of the source.

While the target is attached to the source and the synchronization is taking place, the target remains unavailable to any other host. However, the production host can access the source.

After synchronization is complete, the target can be detached from the source and is made available for BC operations. Figure 13-6 (b) shows full-volume mirroring when the target is detached from the source.

Notice that both the source and the target can be accessed for read and write operations by the productionhosts.

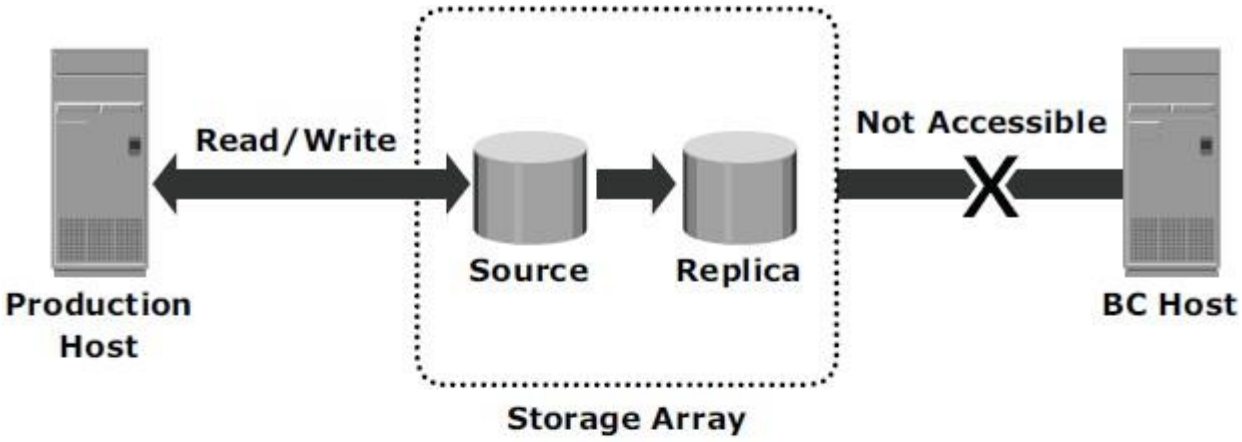
After the split from the source, the target becomes a PIT copy of the source. The point-in-time of a replica is determined by the time when the source is detached from the target. For example, if the time of detachment is 4:00 pm, the PIT for the target is 4:00pm.

After detachment, changes made to both source and replica can be tracked at some predefined granularity. This enables incremental resynchronization (source to target) or incremental restore (target to source).

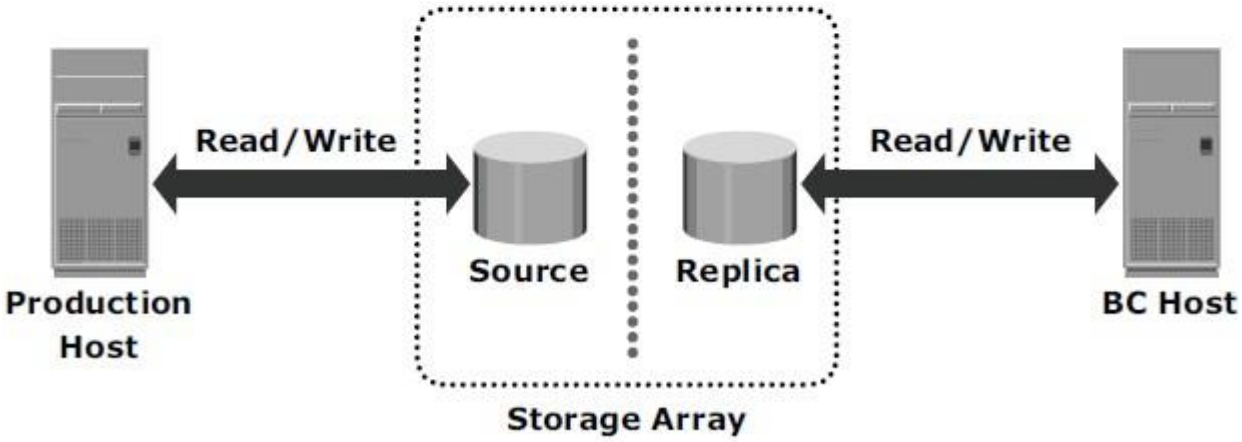
The granularity of the data change can range from 512 byte blocks to 64 KB blocks. Changes are typically tracked using bitmaps, with one bit assigned for each block.

If any updates occur to a particular block, the whole block is marked as changed, regardless of the size of the actual update. However, for resynchronization (or restore), only the changed blocks have to be copied, eliminating the need for a full synchronization (or restore) operation.

This method reduces the time required for theseoperations considerably. In full-volume mirroring, the target is inaccessible for the duration of the synchronization process, until detachment from the source. For large databases, this can take a long time.



(a) Full volume mirroring with source attached to replica



(b) Full volume mirroring with source detached from replica

Figure 13-6: Full-volume mirroring

Pointer-Based, Full-Volume Replication

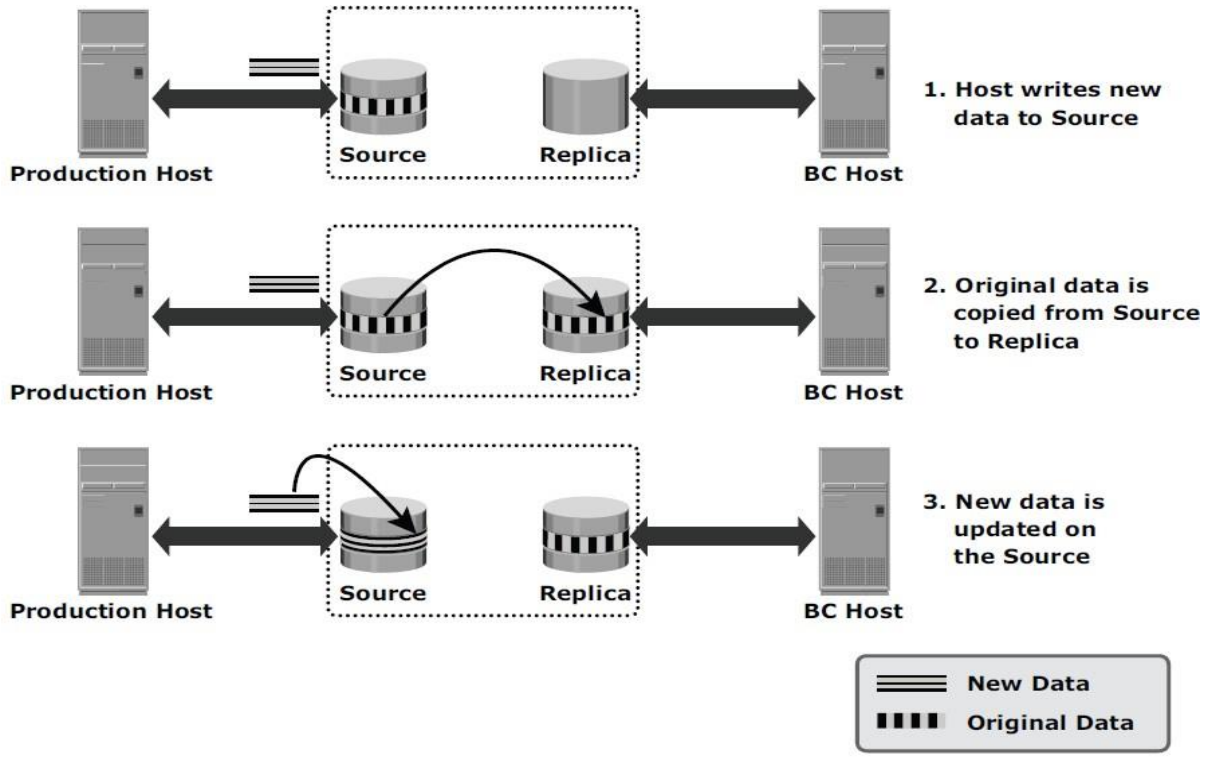
An alternative to full-volume mirroring is *pointer-based full-volume replication*. Like full-volume mirroring, this technology can provide full copies of the source data on the targets. Unlike full-volume mirroring, the target is made immediately available at the activation of the replication session. Hence, one need not wait for data synchronization to, and detachment of, the target in order to access it. The time of activation defines the PIT copy of 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. Pointers are initialized to map the (currently) empty data blocks on the target to the corresponding original data blocks on the source. The granularity can range from 512 byte blocks to 64 KB blocks or higher. Data is then copied from the source to the target, based on the mode of activation.

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

1. A write operation is issued to a specific address on the source for the first time shown in Fig13- 7).
2. A read or write operation is issued to a specific address on the target for the first time shown in Figure 13-8 and Figure13-9.

When a write is issued to the source for the first time after session activation, original data at that address is copied to the target. After this operation, the new data is updated on the source. This ensures that



igure 13-7: Copy on first access (CoFA) — write to source

original data at the point-in-time of activation is preserved on the target. This is illustrated in Figure 13-7.

When a read is issued to the target for the first time after session activation, the original data is copied from the source to the target and is made available to the host. This is illustrated in Figure 13-8.

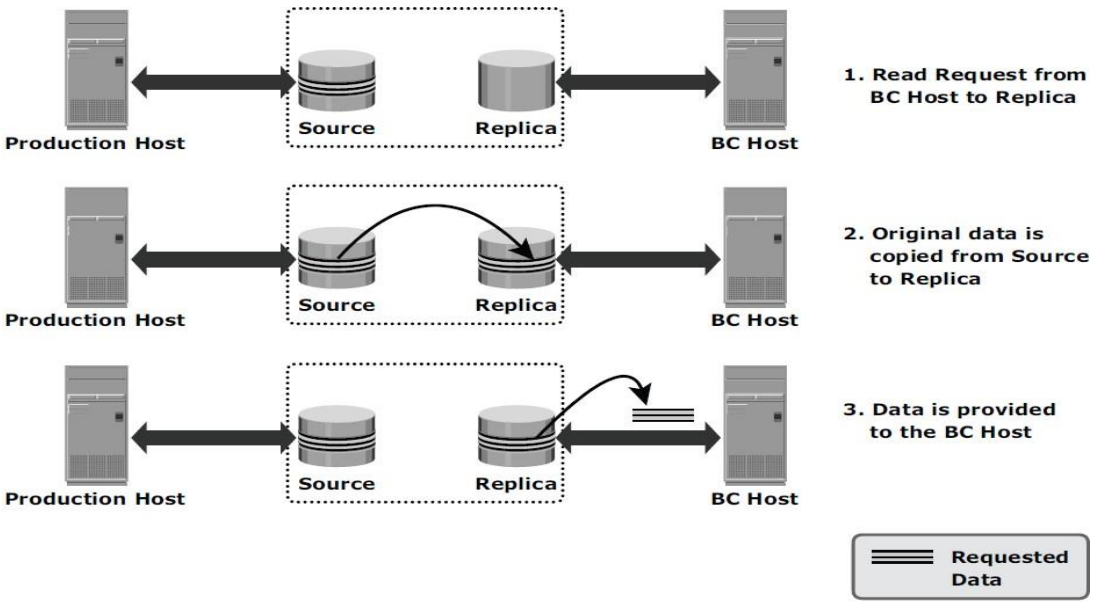


Figure 13-8: Copy on first access (CoFA) — read from target

When a write is issued to the target for the first time after session activation, the original data is copied from the source to the target. After this, the new data is updated on the target. This is illustrated in Figure

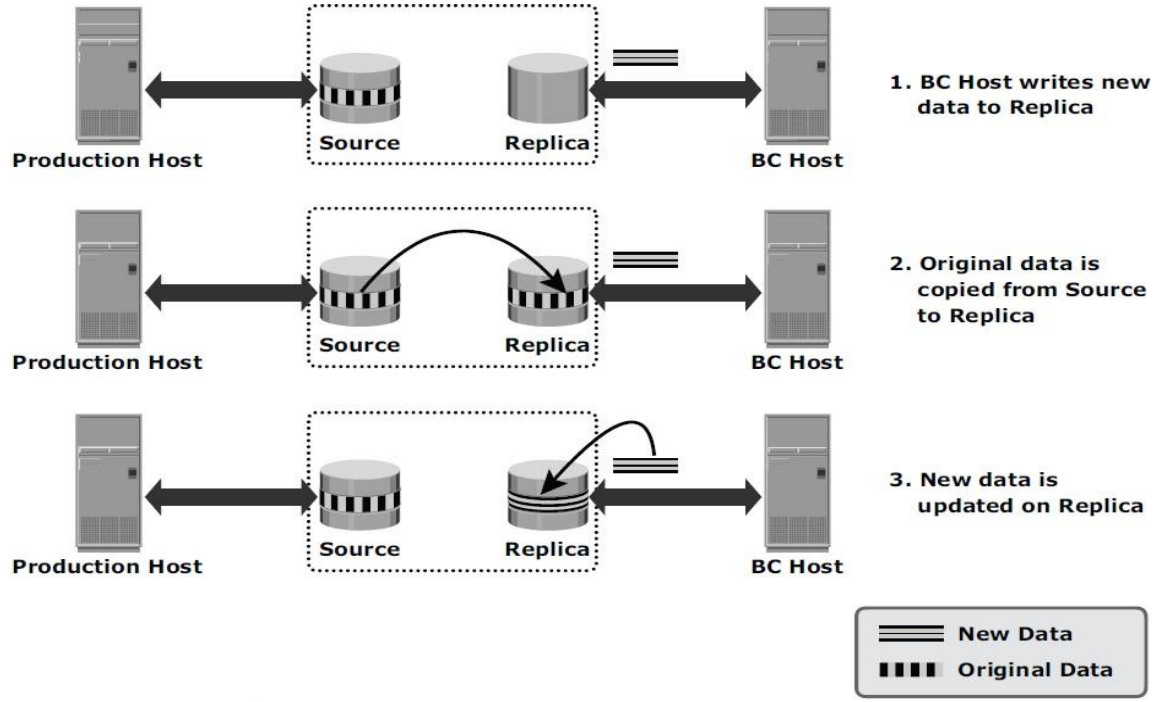


Figure 13-9: Copy on first access (CoFA) — write to target

13-9.

In all cases, the protection bit for that block 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 reads or writes to the same data blocks on the target, do not trigger a copy operation (and hence are termed Copy on First Access).

Pointer-Based Virtual Replication

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

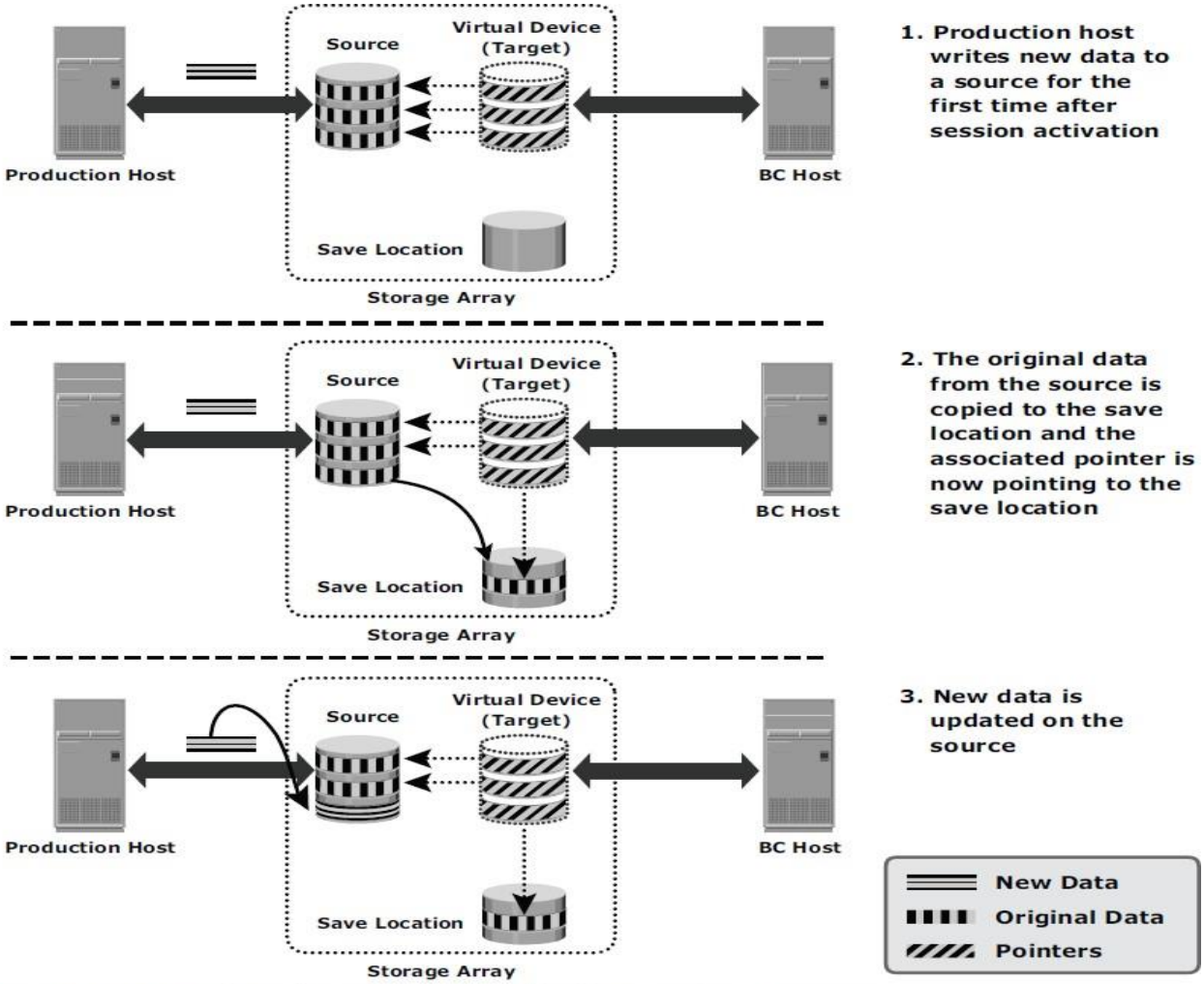


Figure 13-10: Pointer-based virtual replication — write to source

activation, original data at that address is copied to a predefined area in the array. This area is generally termed the *save location*. The pointer in the target is updated to point to this data address in the save location. After this, the new write is updated on the source. This process is illustrated in Figure 13-10.

When a write is issued to the target for the first time after session activation, original data is copied from the source to the save location and similarly the pointer is updated to data in save location. Another copy of the original data is created in the save location before the new write is updated on the save location. This process is illustrated in Figure 13-11.

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

Pointer-based virtual replication uses CoFW technology. Subsequent writes to the same data block on the source or the target do not trigger a copy operation.

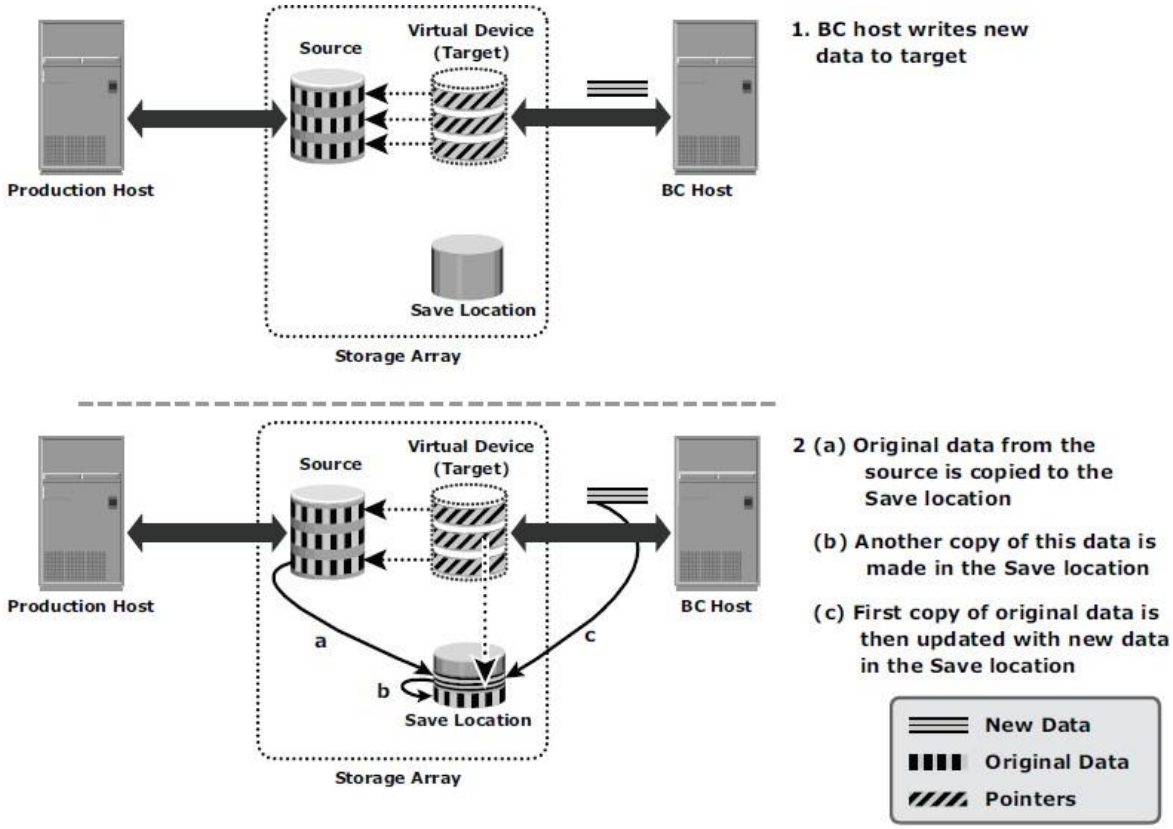


Figure 13-11: Pointer-based virtual replication – write to target

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. As the target only contains pointers to data, 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 expected data change.

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 heteroge- neous environments. Continuous data protection (CDP) is a technology used for network-based local and remote replications.

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 opera- tion, 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 implementa- tion 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.

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.

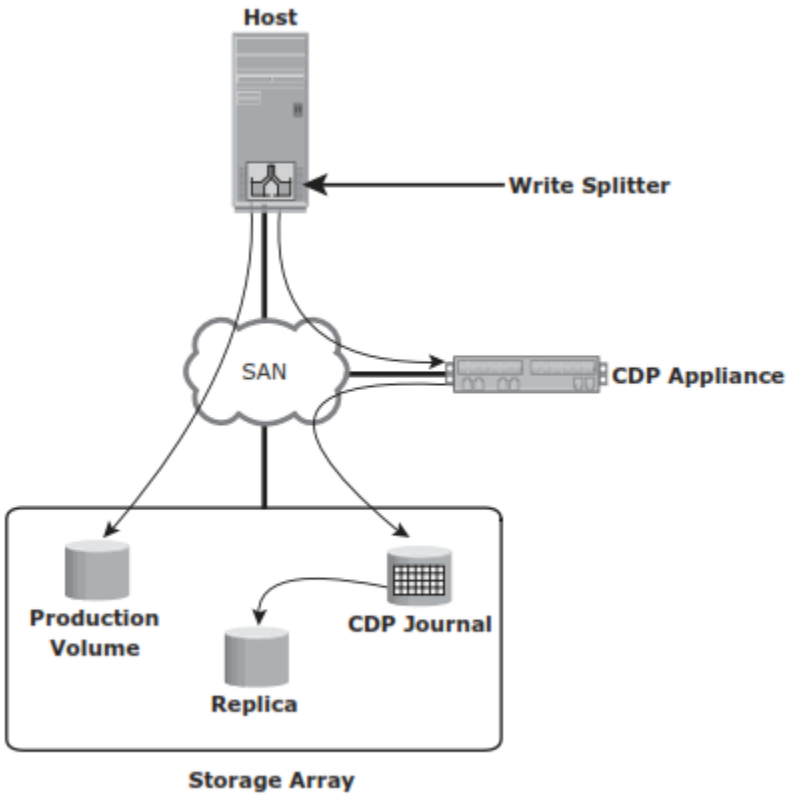


Figure 11-14: Continuous data protection – local replication

Restore and Restart Considerations

Local replicas can be used to restore data to production devices. Alternatively, applications can be restarted using the consistent point-in-time copy of the data on the replicas.

A replica can be used to restore data to the production devices in the event of logical corruption of production devices — i.e., the devices are available but the data on them is invalid. Examples of logical corruption include accidental deletion of information (tables or entries in a database), incorrect data entry, and incorrect updating to existing information. Restore operations from a replica are incremental and provide a very small RTO. In some instances, applications can be resumed on the production devices prior to completion of the data copy. Prior to the restore operation, access to production and replica devices should bestopped.

Production devices may also become unavailable due to physical failures, such as production server or physical drive failure. In this case, applications can be restarted using data on the latest replica. If the production server fails, once the issue has been resolved, the latest information from the replica devices can be restored back to the production devices. If the production device(s) fail, applications can

continue to run on replica devices. A new PIT copy of the replica devices can be created or the latest information from the replica devices can be restored to a new set of production devices. Prior to restarting applications using the replica devices, access to the replica devices should be stopped. 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 replicadevices.

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 a full-volume copyoperation.

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 original source volumes. If the original source volume is inaccessible for any reason, these replicas cannot be used for a restore or a restart. Table 13-1 presents a comparative analysis of the various storage array–based replicationtechnologies.

Table 13-1: Comparison of Local Replication Technologies

FACTOR	FULL-VOLUME MIRRORING	POINTER-BASED, FULL-VOLUME REPLICATION	POINTER-BASED VIRTUAL REPLICATION
Performance impact on source	No impact	CoFA mode - some impact Full copy - 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
Accessibility of source for restoration	Not required	CoFA mode - required Full copy - not required	Required
Accessibility to target	Only after synchro-nization and detach-ment from the source	Immediately accessible	Immediately accessible

Tracking Changes to Source and Target

Updates occur on the source device after the creation of point-in-time local replicas. If the primary purpose of local replication is to have a viable pointin- time copy for data recovery or restore operations, then the target devices should not be modified. Changes can occur on the target device if it is used for non-BC operations. To enable incremental resynchronization or restore operations, changes to both the source and target devices after the point-in-time can be tracked. This is typically done using bitmaps, with one bit per block of data. The 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. The size of the bitmap would be 4 KB. If any or all of a 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 target are then flagged by setting the appropriate bits to 1 in the bitmap. When resynchronization or a restore is required, a *logical OR* operation between the source bitmap and the target bitmap is performed. The bitmap resulting from this operation (see Figure 13-12) references all blocks that have been modified in either the source or thetarget.

This enables an optimized resynchronization or a restore operation, as it eliminates the need to copy all the blocks between the source and the target. The direction of data movement depends on whether a resynchronization or a restore operation is performed.

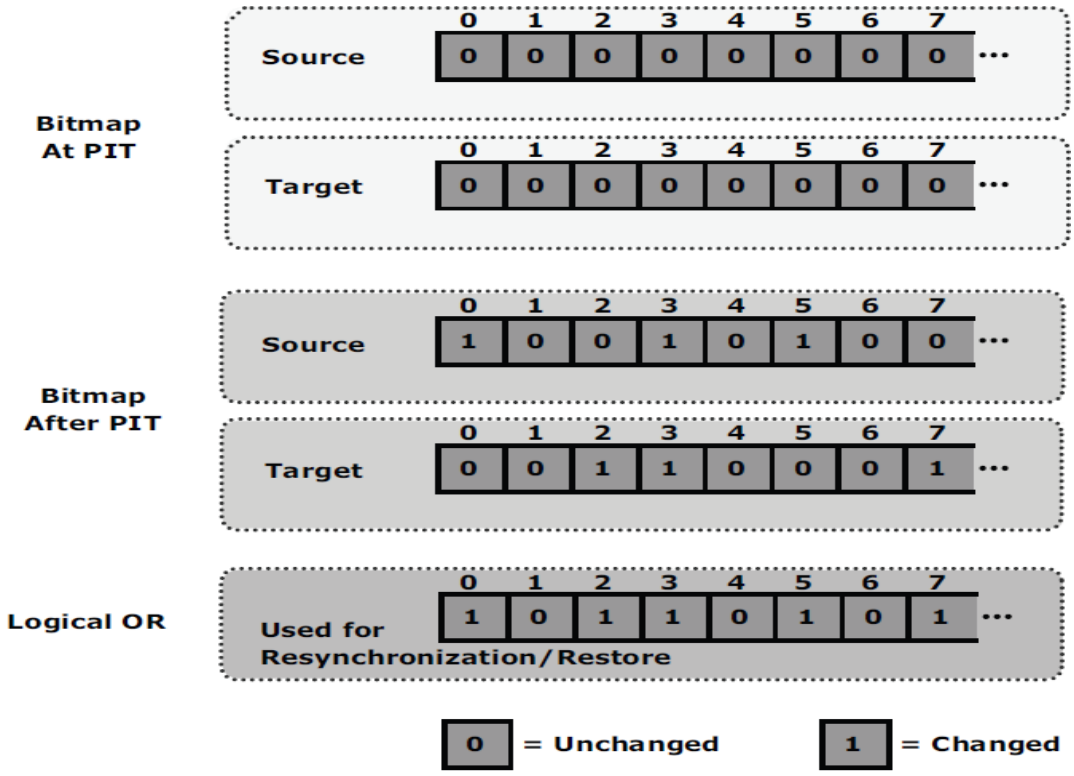


Figure 13-12: Tracking changes

If resynchronization is required, then changes to the target are overwritten with the corresponding blocks from the source. In this example, that would be blocks 3, 4, and 8 on the target (from the left).

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

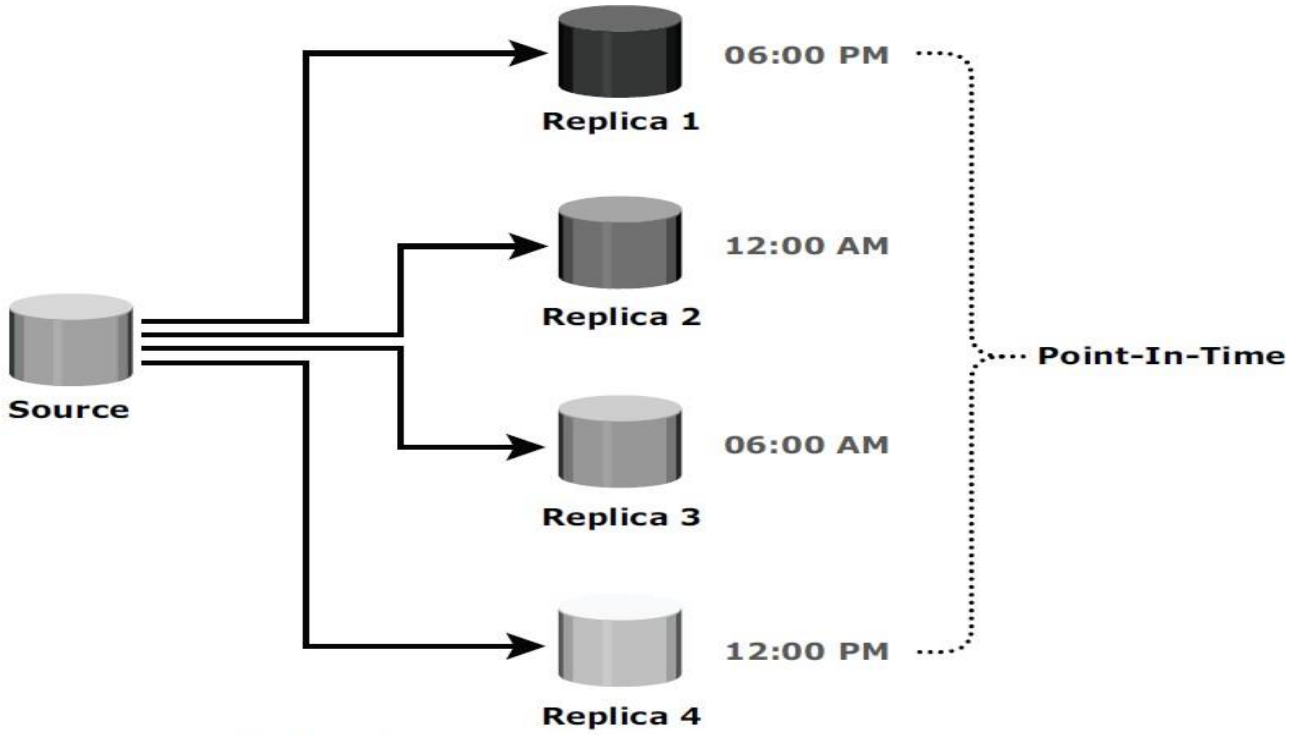


Figure 13-13: Multiple replicas created at different points in time

activities and as a restore point. Figure 13-13 shows an example in which a copy is created every six hours from the samesource.

If the source is corrupted, the data can be restored from the latest PIT copy. The maximum RPO in the example shown in Figure 13-15 is six hours. More frequent replicas will reduce the RPO and the RTO.

Management Interface

The replication management software residing on the storage array provides an interface for smooth and error-free replication. This management software interface provides options for synchronization, resynchronization, splitting, starting and stopping a replication session, and monitoring replication performance. In general, two types of interface are provided:

1. **CLI:** An administrator can directly enter commands against a prompt in a CLI. A user can also perform some command operations based on assigned privileges. CLI scripts are developed to perform specific BC operations in a database or applicationenvironment.

2. **GUI:** This type of interface includes the toolbar and menu bar options. A user can select an operation name and monitor performance in real time. Most of the tools have a browser-based GUI to transfer commands and are often integrated with other storage management suite products.

Local Replication in a Virtualized 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 applica- tions on multiple VMs is a time-consuming task; VM Clone helps to simplify this process.

1. A database is stored on ten 9-GB RAID 1 LUNs. A cascade three-site remote replication solution involving a synchronous and disk-buffered solution has been chosen for disaster recovery. All the LUNs involved in the solution have RAID 1 protection. Calculate the total amount of raw capacity required for this solution.

   Given:

   RAID type = RAID 1

   Drive Capacity (GB) = 9

   Number of drives in a RAID group = 2 Number of RAID groups = 10

   Therefore.

   The total amount of raw capacity: For 1 group

   1 drive has 9GB

   2 drives has 18GB

   1 group = 18 GB

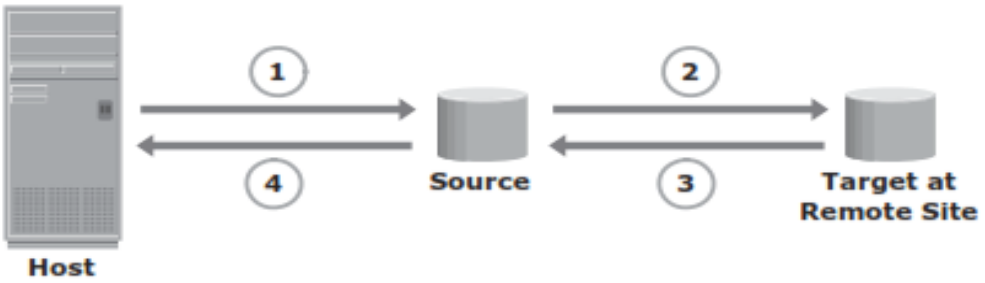
   For 10 groups = 18 x 10 GB= 180 GB

Remote Replication

- Remote replication is the process to createreplicas of information assets at remotesites (locations).
- Remote replication helpsorganizations mitigate the risks associated withregionally driven outages resulting from naturalor human-made disasters.
- During disasters, theworkload can be moved to a remote site to ensurecontinuous business operation. Similar to localreplicas, remote replicas can also be used for otherbusiness operations.

Modes of Remote Replication

- The two basic modes of remote replication are synchronous and asynchro- nous. 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 receivedat 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, band- width, 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 fromseveral 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.

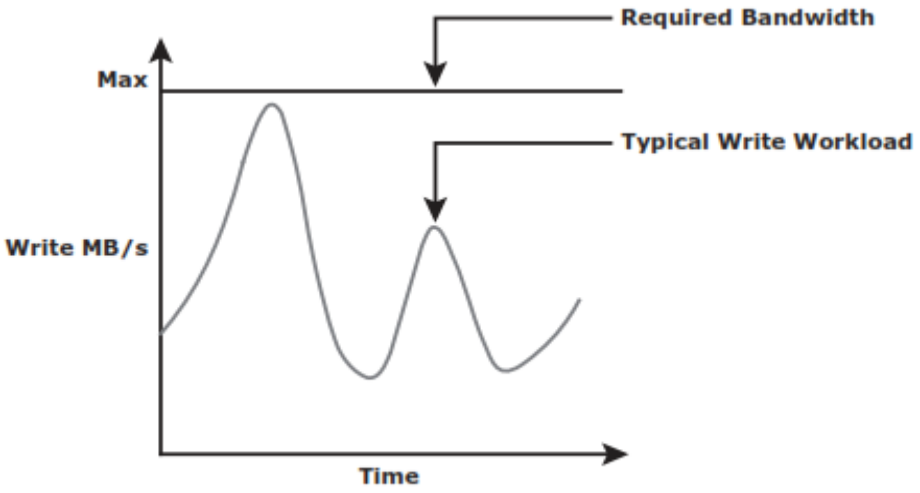
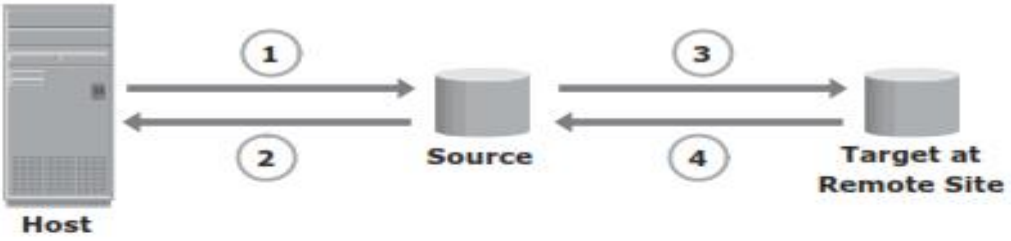


Figure 12-2: Bandwidth requirement for synchronous replication



1. The host writes data to the source.
2. The write is immediately acknowledged to the host.
3. Data is transmitted to the target at a remote site later.
4. The target acknowledges back 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.

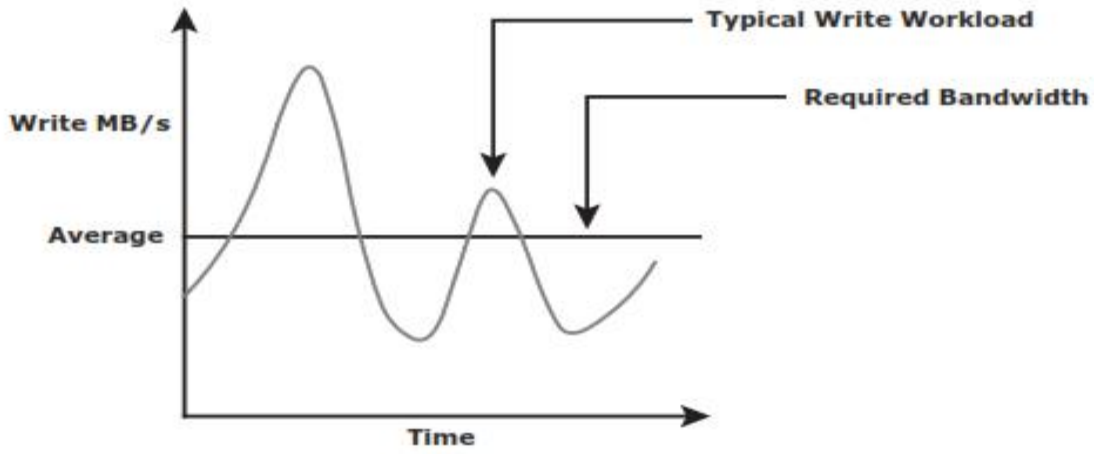


Figure 12-4: Bandwidth requirement for asynchronous replication

- 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.

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 synchro- nization 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).

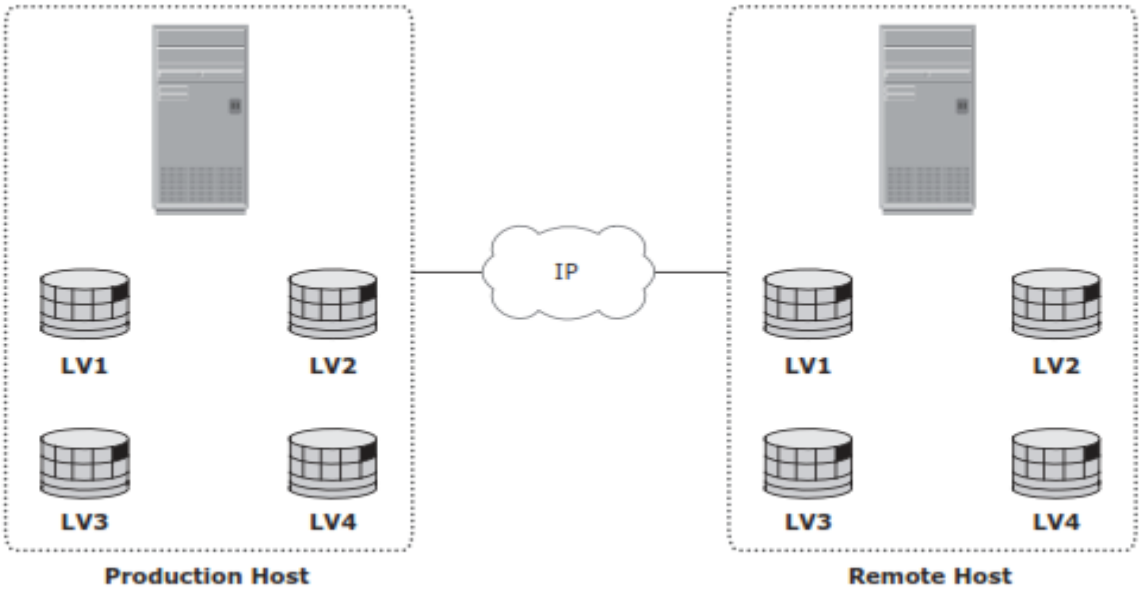


Figure 12-5: LVM-based remote replication

- 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 there- fore supports replication between heterogeneous storage arrays. Most operatingsystems 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.

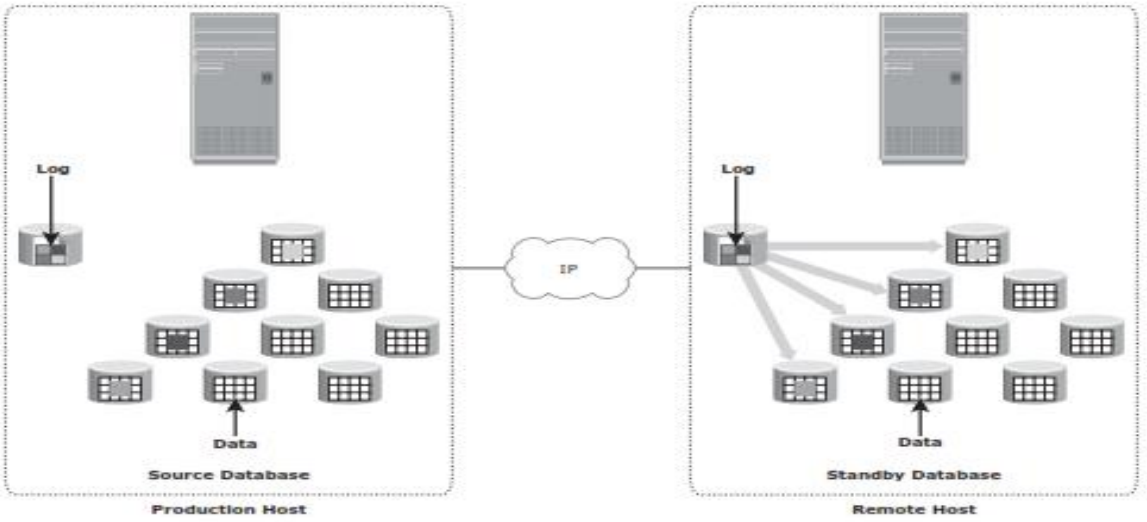


Figure 12-6: Host-based log shipping

- 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

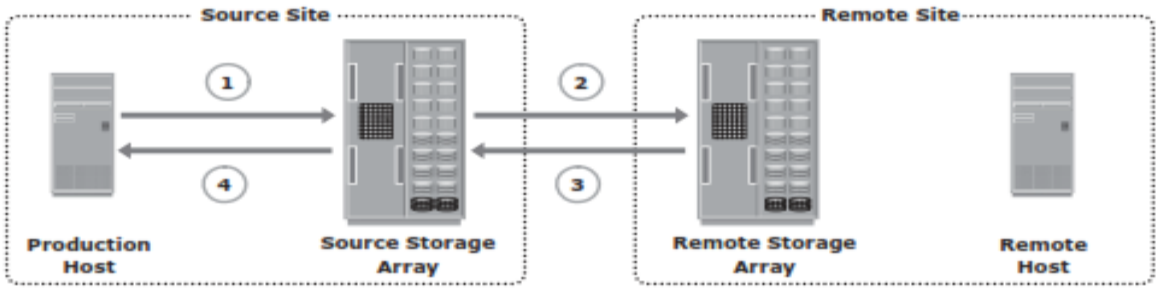
- In storage array-based remote replication, the array-operating environment and

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 produc- tion 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 out- age, if there is a failure at the source site, some data will be lost, and the RPO at the target will not be zero.

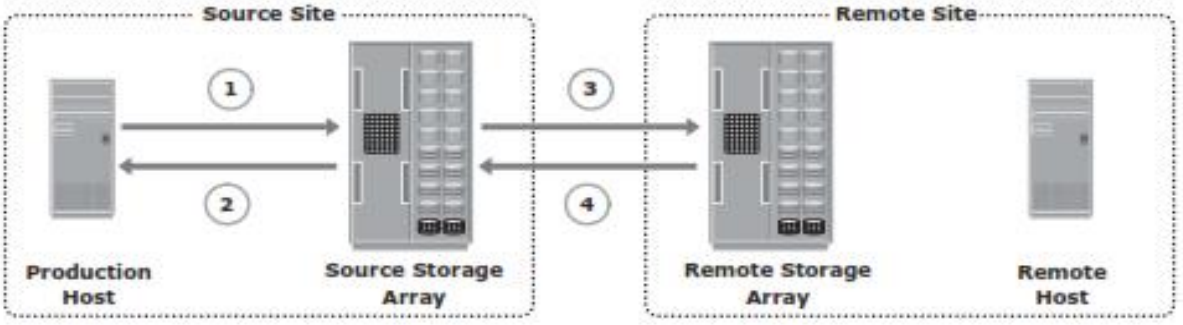


1. Write from the production host is received by the source storage array.
2. Write is then transmitted to the remote storage array.
3. Acknowledgment is sent to the source storage array by the remote storage array.
4. Source storage array signals write-completion to the production host.

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 replica- tion. 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 theorder 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.

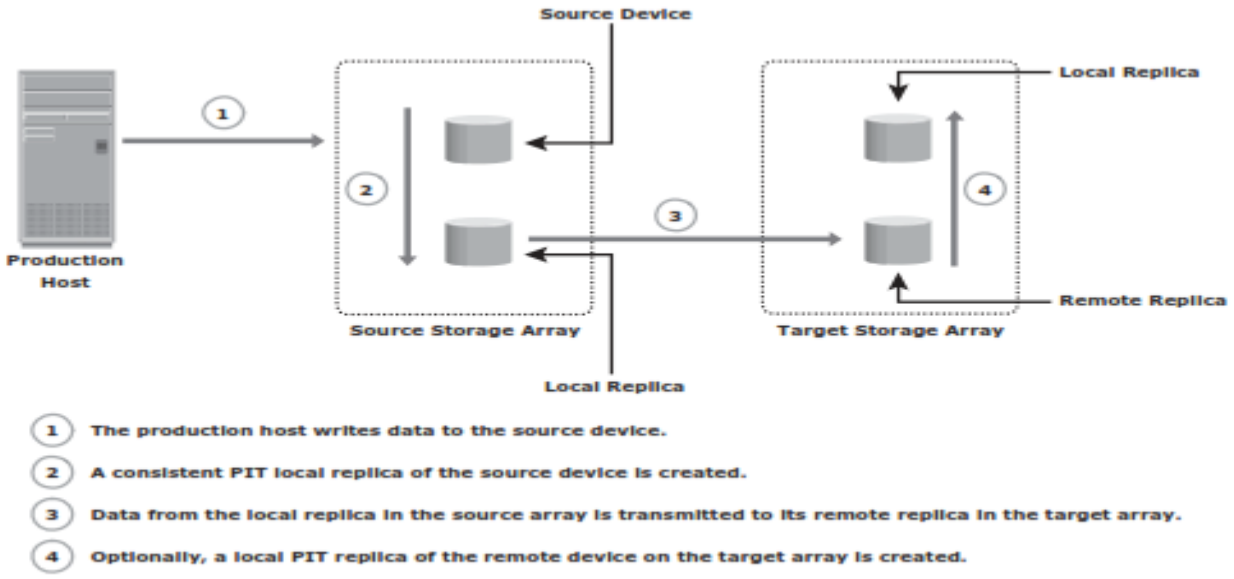


Figure 12-9: Disk-buffered remote replication

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.

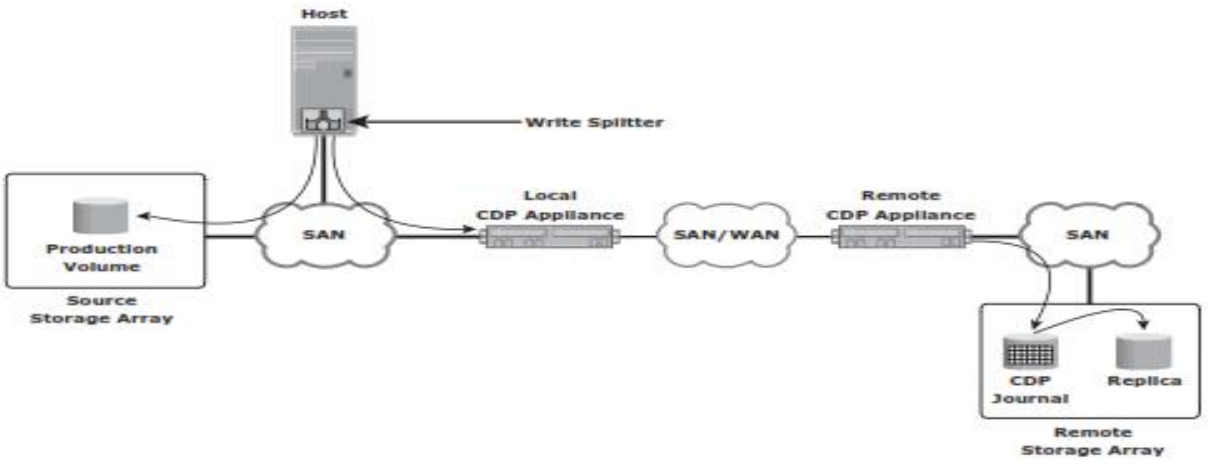


Figure 12-10: CDP remote replication

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

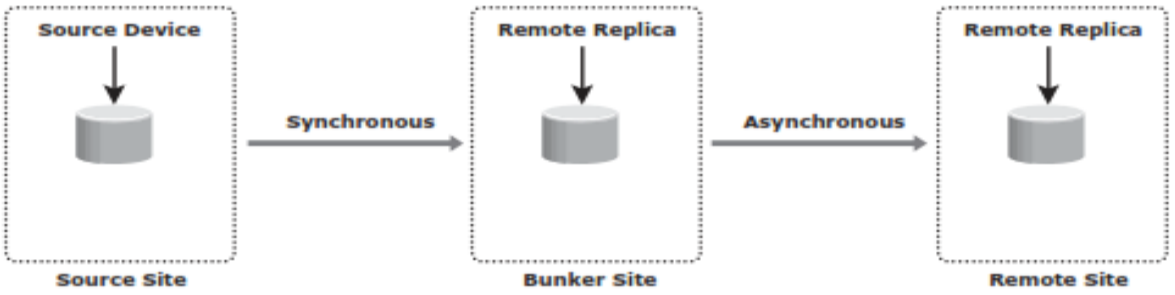
In the cascade/multihop three-site replication, data flows from the source to the intermediate storage array, known as a bunker, in the first hop, and then from a bunker to a storage array at a remote site in the second hop.

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

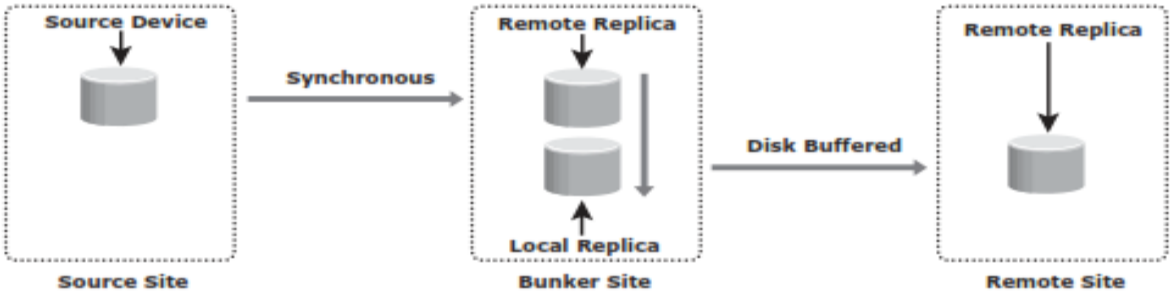
synchronously, but replication between the bunker and the remote site can be achieved either as disk-buffered mode or asynchronous mode.

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.



(a) Synchronous + Asynchronous



(b) Synchronous + Disk Buffered

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 can- not 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 dur- ing 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 asyn- chronous 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.

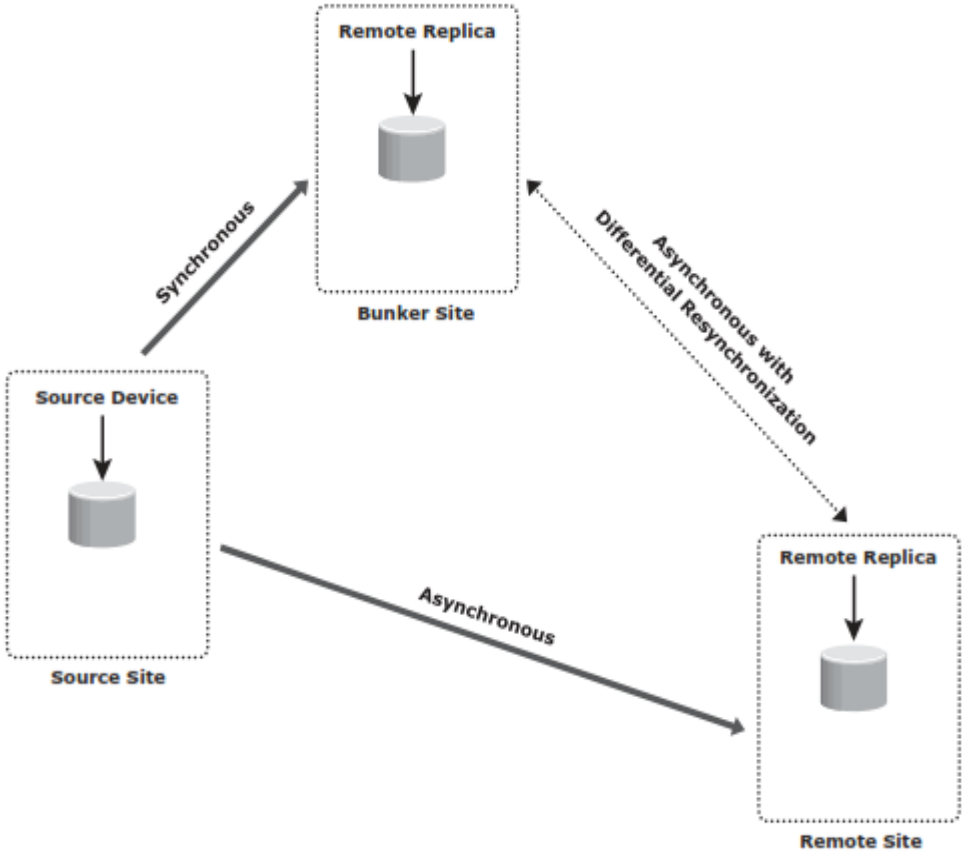


Figure 12-12: Three-site replication triangle/multitarget

- 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.

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.

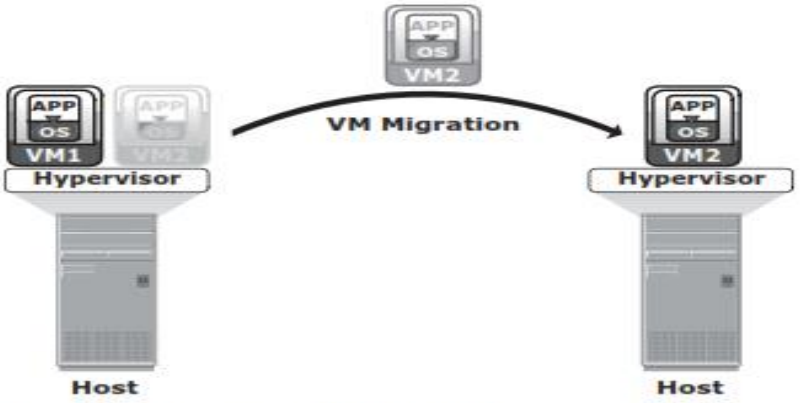


Figure 12-14: Hypervisor-to-hypervisor VM migration

- In array-to-array VM migration, virtual disks are moved from the source array to the remote array.
- This approach enables the administrator to moveVMs 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.

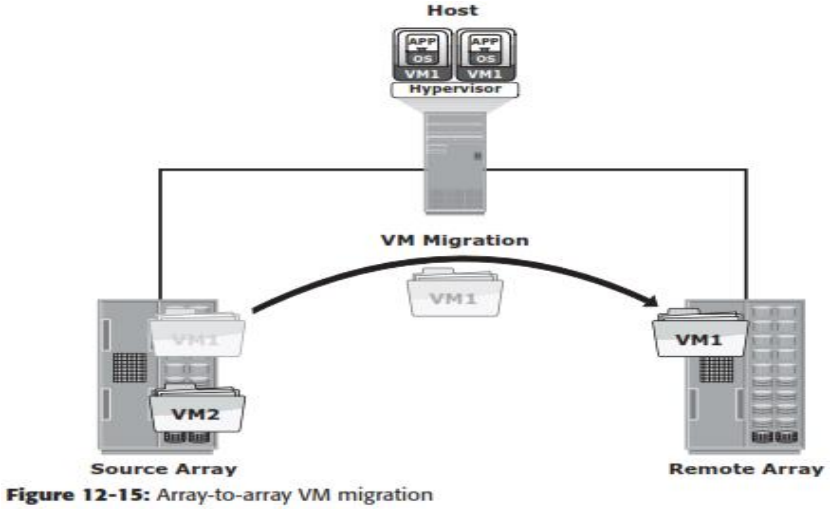


Figure 12-15: Array-to-array VM migration

SECURING AND MANAGING STORAGE INFRASTRUCTURE

5.1 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.
- **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. 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.

5.2 Risk Triad

Risk triad defines risk in terms of threats, assets, and vulnerabilities. They are considered from the perspective of risk identification and control analysis.

5.2.1 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

5.2.2 Security Threats

- Threats are the potential attacks that can be carried out on an IT infrastructure.
- 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. **Denial of service (DoS**) attacks prevent legitimate users from accessing resources and services. **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.

5.2.3 Vulnerabilities

- 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.
- 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.
    - An **Attack surface** refers to the various entry points that an attacker can use to launch an attack.
    - An **attack vector** is a step or a series of steps necessary to complete an attack.
    - **Work factor** refers to the amount of time and effort required to exploit an attack vector.

5.3 Storage Security Domains

- 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.
- Fig 5.1 depicts the three security domains of a storage system environment.

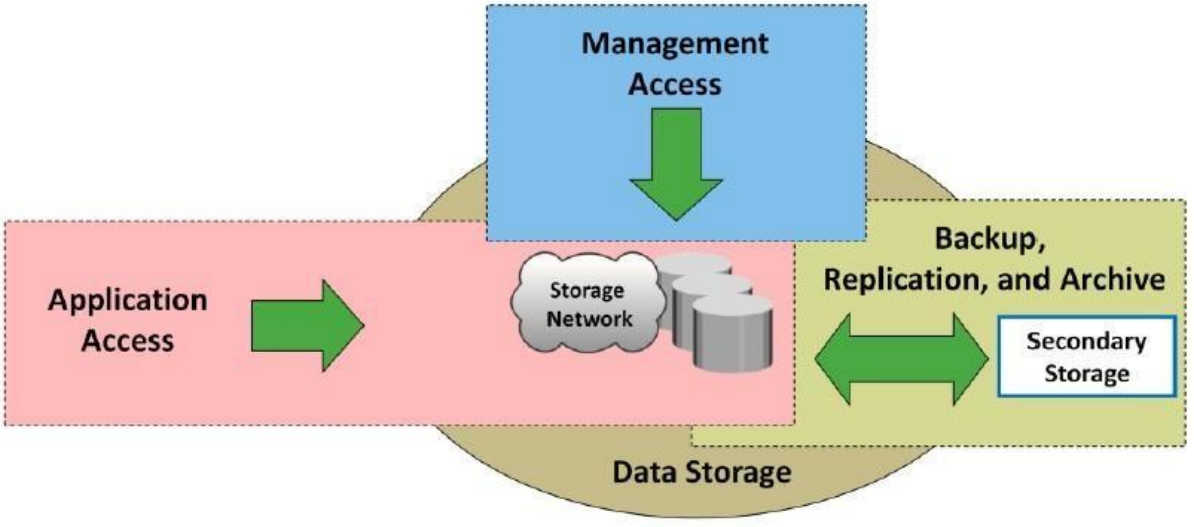


Fig 5.1: Storage security domains

- 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

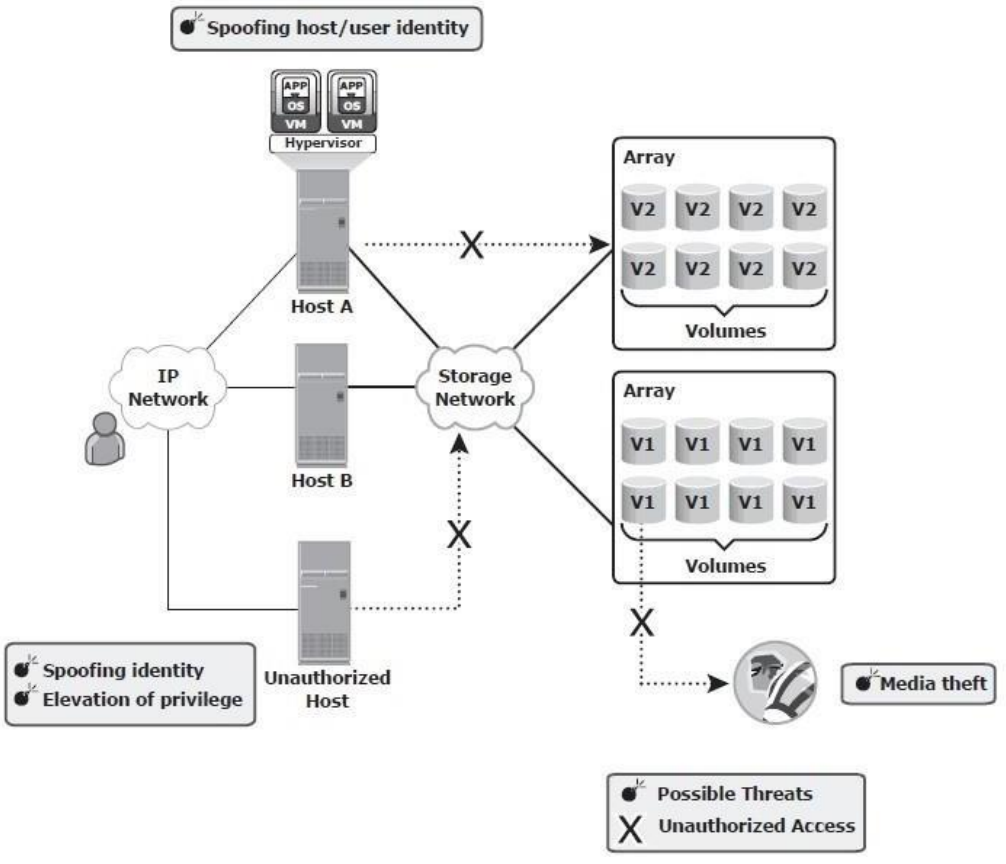
devices and to the data residing on those devices. This domain is primarily accessed by storage administrators who configure and manage the environment.

- 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.

5.3.1 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.
- Fig 5.2 shows application access in a storage networking environment.

Fig 5.2: Security threats in an application access domain



- 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.

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 net- work 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 spoof- ing. 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 con- trols 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 log- ging 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.

5.3.2 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.
- Fig 5.3 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. 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.
- 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.

- 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.

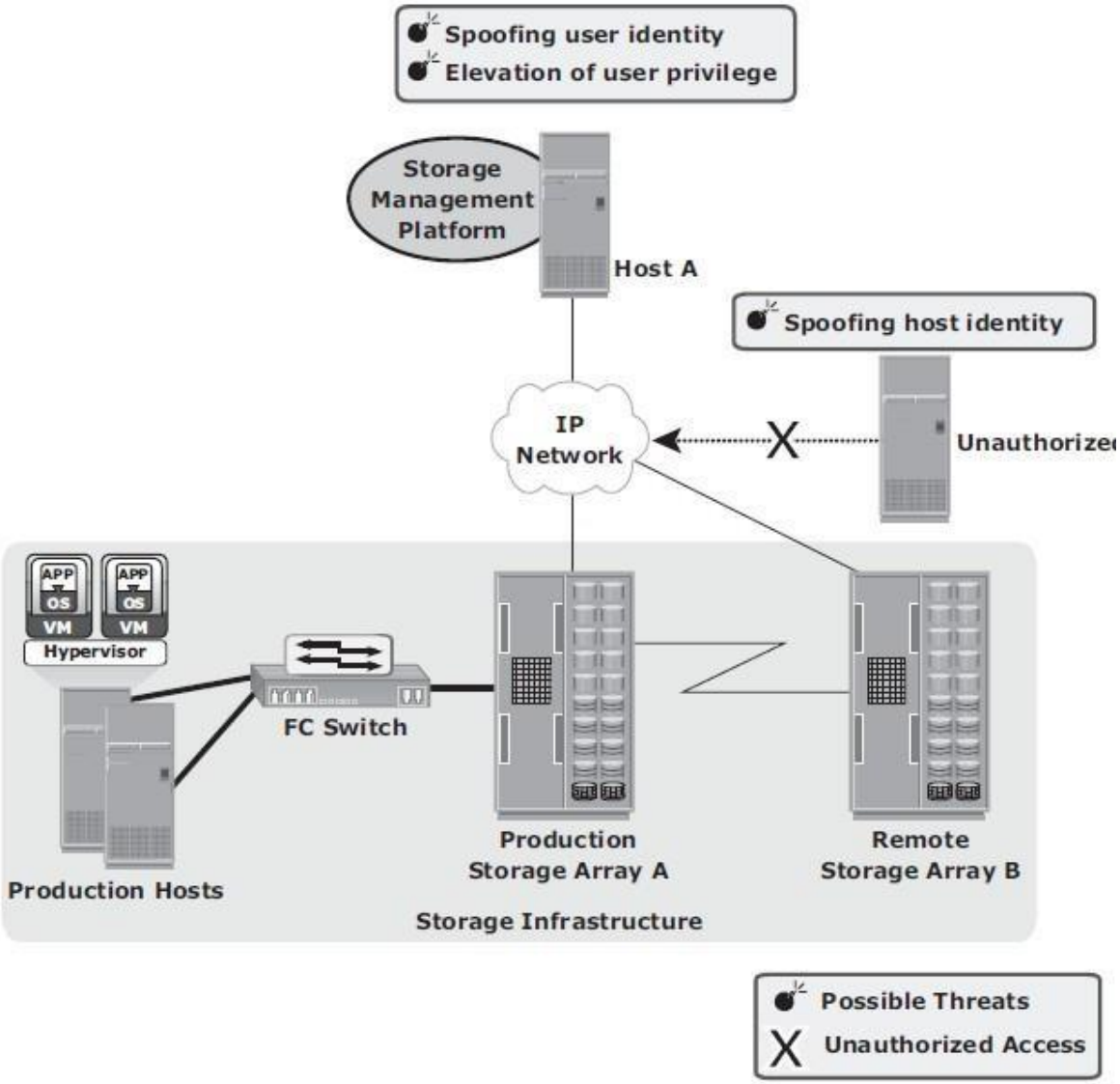


Fig 5.3: Security threats in a management access domain

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.

5.3.3 Securing Backup, Replication and Archive

- Backup, replication, and archive is the third domain that needs to be secured against an attack.
- 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.
- Fig 5.4 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.
- The physical threat of a backup tape being lost, stolen, or misplaced, especially if the tapes contain highly confi dential 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.

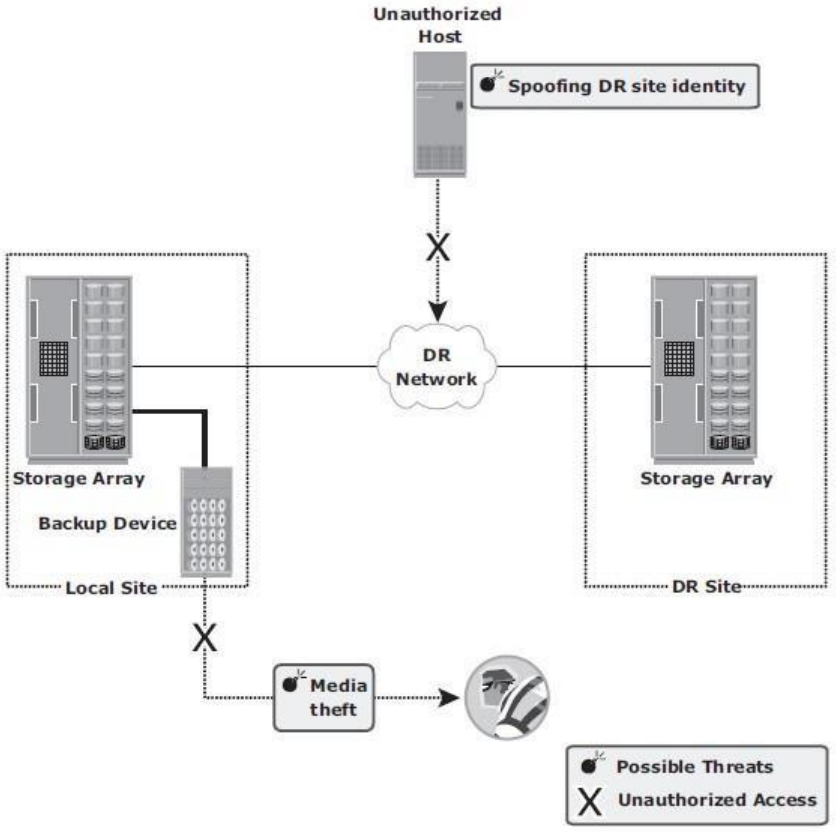


Fig 5.4: Security threats in a backup, replication, and archive environment

5.4 Security solutions for FC-SAN, IP-SAN, NAS Environment

5.4.1 FC-SAN

- Traditional FC SANs have 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.

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.
- Fig 5.5 illustrates various levels (zones) of a storage networking environment that must be secured and the security measures that can be deployed.

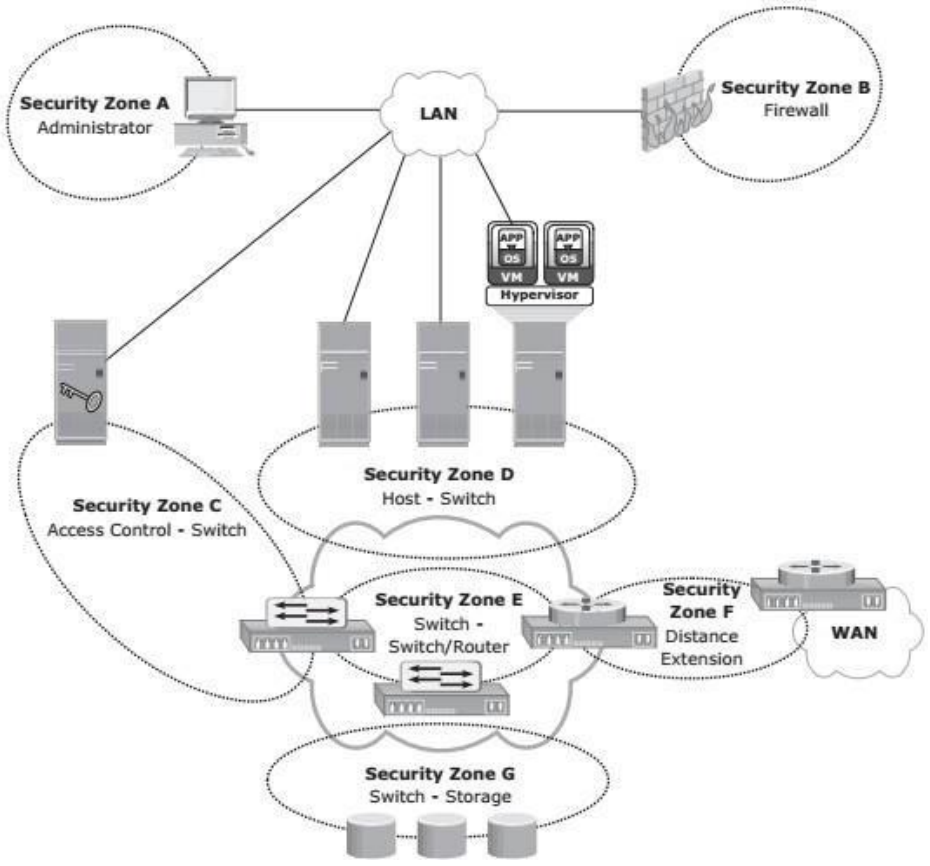


Fig 5.5: FC SAN security architecture

- Table 5.1 provides a comprehensive list of protection strategies that must be implemented in various security zones. Some of the security mechanisms listed in Table 5.1 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.

Table 5.1 : list of 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.
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.

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.

LUN Masking and Zoning

- LUN masking and zoning are the basic SAN security mechanisms used to protect against unauthorized access to storage.
- The standard implementations of LUN masking on storage arrays mask the LUNs presented to a frontend 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 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.

- Access control lists (ACLs)
  - Include device connection and switch connection control policies
    - Device connection control policy specifies which HBAs, storage ports can be connected to a particular switch
    - Switch connection control policy prevents unauthorized switches to join a particular switch
- Fabric Binding
  - Prevents unauthorized switch from joining a fabric
- Role-based access control (RBAC)
  - Enables assigning roles to users that explicitly specify access rights

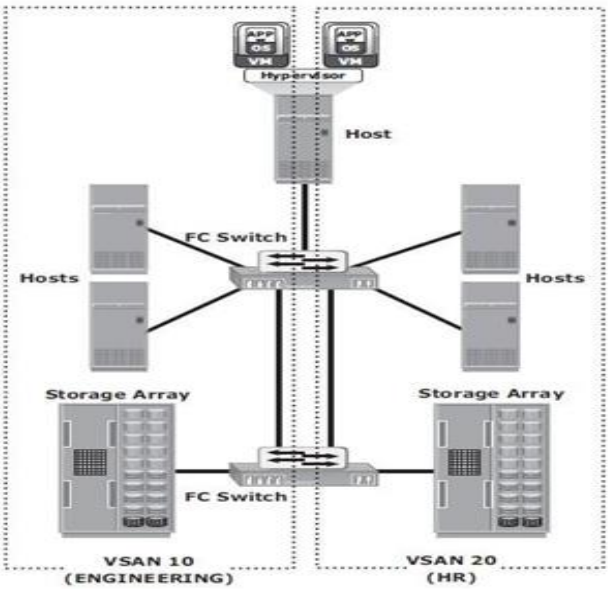


Fig 5.6: Securing SAN with VSAN

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.

  Fig 5.6 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.

5.4.2 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.
- 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)
  - 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. 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 fi le 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.
- Fig 5.7 depicts the authentication process 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.

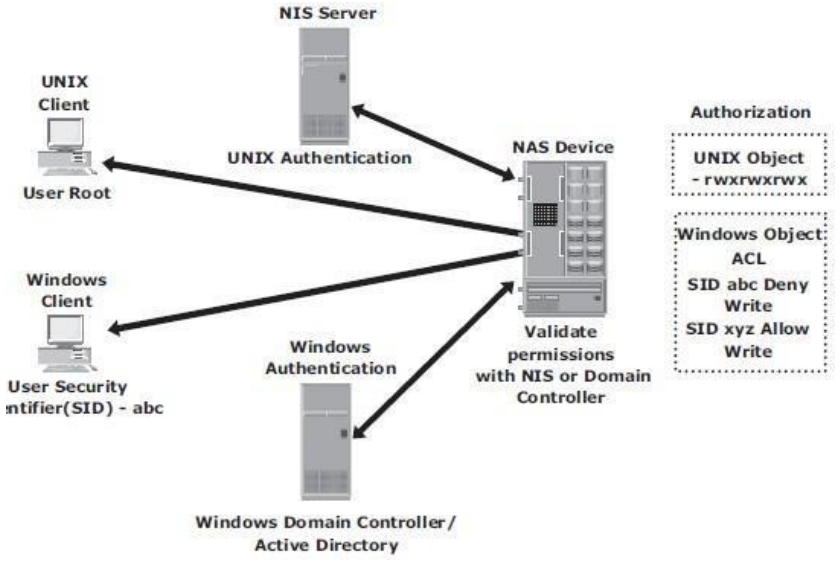


Fig 5.7 Securing user access in a NAS environment

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.
- 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 Fig 5.8 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.

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 key tab 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.

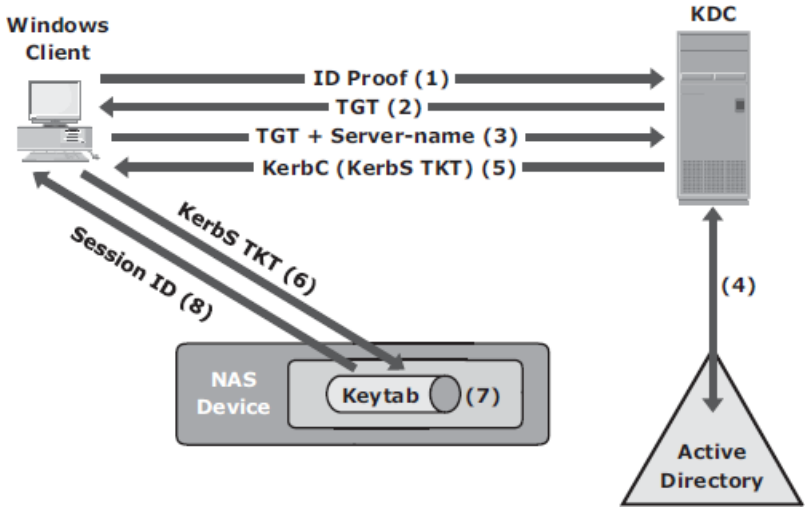


Fig 5.8 Kerberos authorization

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 confi ed 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.
- Fig 5.9 depicts a typical firewall implementation.

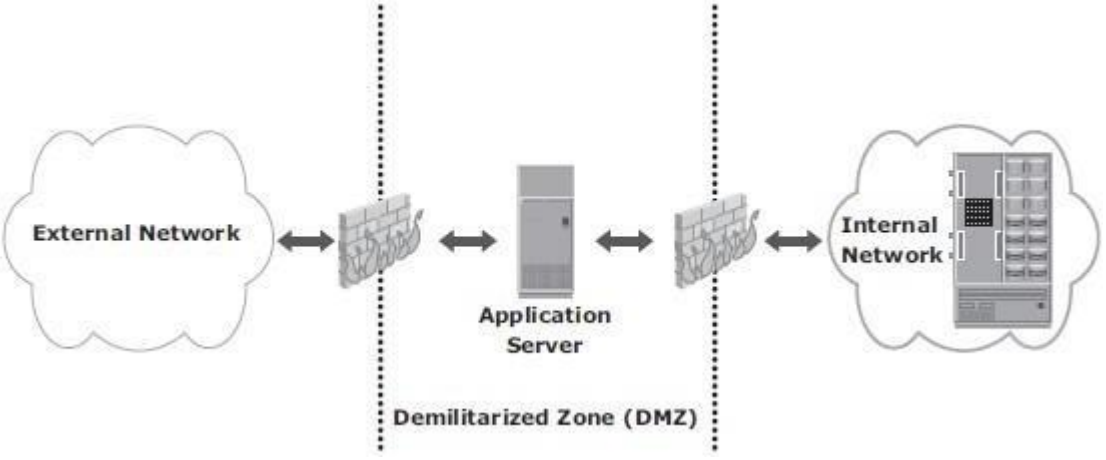


Fig 5.9: Securing a NAS environment with a network-layer firewall

- 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. No Internet-based traffic is allowed to penetrate the second set of firewalls and gain access to the internal network. 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.

5.4.3 IP SAN

- The *Challenge-Handshake Authentication Protocol* (CHAP) is a basic authentication mechanism that has been widely adopted by network devices an 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. Fig 5.10 depicts the CHAP authentication process.

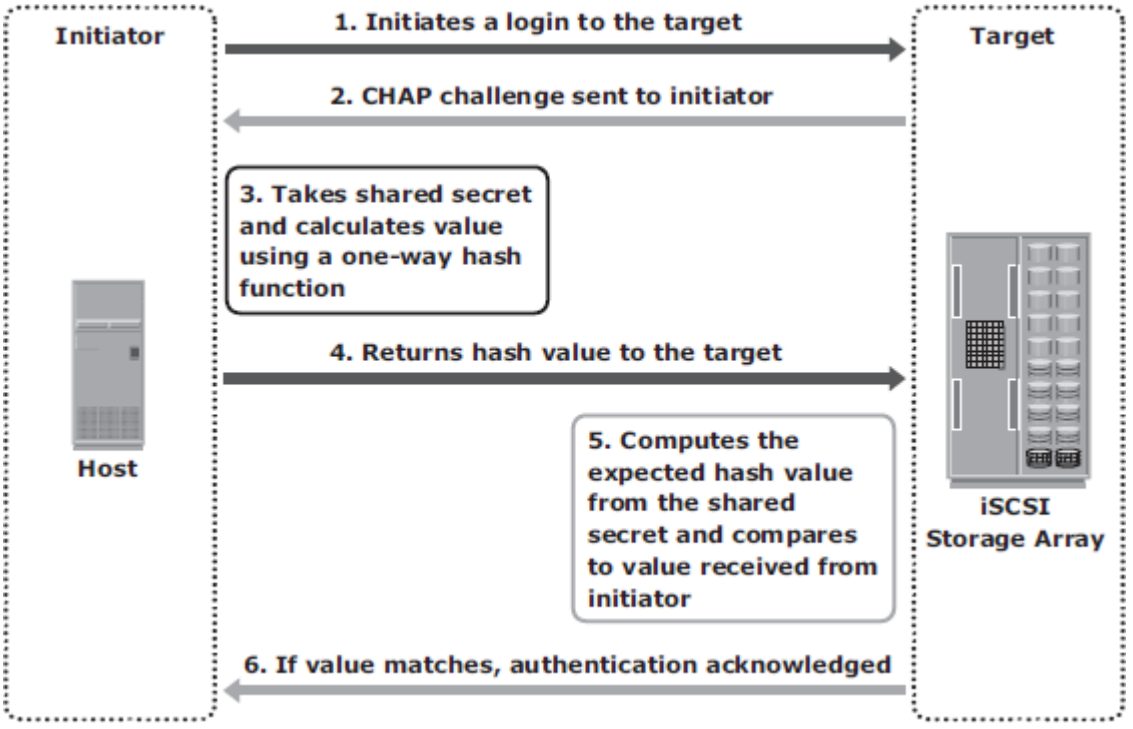


Fig 5.10 : Securing IPSAN with CHAP authentication

- 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 confi gured in the same discovery domain.
- State change notifications (SCNs) inform the iSNS server when devices are added to or removed from a discovery domain. Fig 5.11 depicts the discovery domains in iSNS.

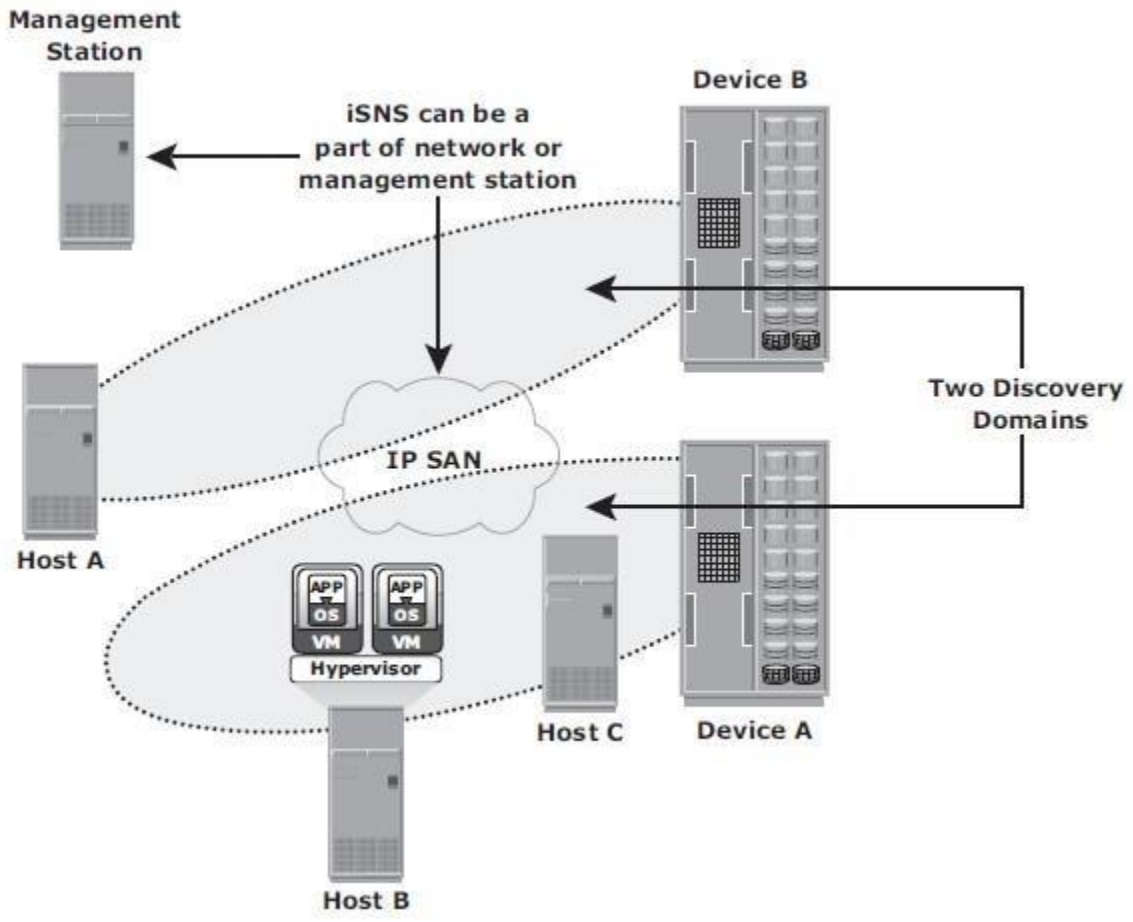


Fig 5.11 : Securing IPSAN with iSNS discovery domains

5.5 Securing Storage Infrastructure in Virtualized & Cloud Environments

5.5.1 Security Concerns

- Organizations are rapidly adopting virtualization and cloud computing, however they have some security concerns.
- The 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.
- **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.

5.5.2 Security Measures

- Security measures can be implemented at the compute, network, and storage levels.

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.
-

- The hypervisor management system must also be protected.
- 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.
- Hardening is a process to change the default configuration to achieve greater security.
- Apart from the measures to secure a hypervisor and VMs, virtualized andncloud 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.
- Intrusion Detection (ID) is the process to detect events that can compromise the confidentiality, integrity, or availability of a resource.

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

5.6 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. Monitoring also helps to analyze the utilization and consumption of various storage infrastructure resources.

5.6.1 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.
- **Capacity** refers to the amount of storage infrastructure resources available.
- **Performance** monitoring evaluates how efficiently different storage infrastructure components are performing and helps to identify bottlenecks.
- **Security** monitoring helps to track unauthorized configuration changes to storage infrastructure resources.

5.6.2 Components Monitored

- The components within the storage environment that should be monitored are:
  - Hosts,
  - Networks, and
  - Storage
- The components are monitored for below parameters:
  - 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 (hardware) failure might cause inaccessibility of the host to its user.

- Server clustering is a mechanism that provides high availability if a server failure occurs.
- ***Capacity monitoring of the file system utilization*** is important to ensure that sufficient capacity is available to the applications, otherwise this disrupts application availability.
- 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.
- ***Performance monitoring of the host*** mainly involves a status check on the utilization of various server resources, such as *CPU* and *memory*.
- High utilization leads to *degraded performance and slower response time*.
- Actions taken by administrators to correct the problem are, *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.
- **Accessibility:** Uninterrupted access to data depends on the accessibility of both the physical and logical components.
- The physical components include **switches, ports, and cables**.
- The logical components include constructs, such as **zones**.
- Any failure in the physical or logical components causes **data unavailability**.
- **Capacity monitoring** in a storage network involves monitoring the number of available ports in the fabric, the utilization of the inters witch links, or individual ports, and each interconnect device in the fabric.

- **Performance monitoring** of the storage network enables assessing individual component performance and helps to identify network bottlenecks.
- For IP networks, monitoring the performance includes monitoring network latency, packet loss, bandwidth utilization for I/O, network errors, packet retransmission rates, and collisions.
- **Security monitoring** of storage network provides information about any unauthorized change to the configuration of the fabric.
- 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 configured with redundant hardware components, and therefore individual component failure does not affect their accessibility.
- Failure of any process in the storage array might disrupt or compromise business operations. Example: 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.
- **Performance monitoring** of a storage array involves using a number of performance metrics, such as utilization rates of the various storage array components, I/O response time, and cache utilization.
- 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.

5.6.3 Monitoring

Examples

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 Fig 5.12.
- All the servers share two storage ports on the storage array and multipathing software is installed on all the servers.
- 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.
- 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.

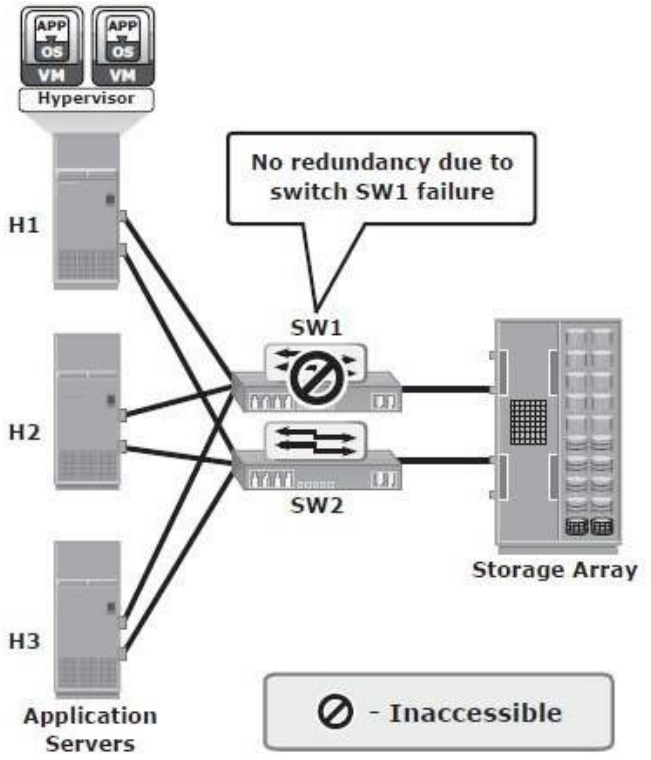


Fig 5.12: Switch failure in a storage infrastructure

Capacity Monitoring

- In the scenario shown in Fig 5.13, 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 on the array helps to 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 decide to whether the new server can be connected to the switches.

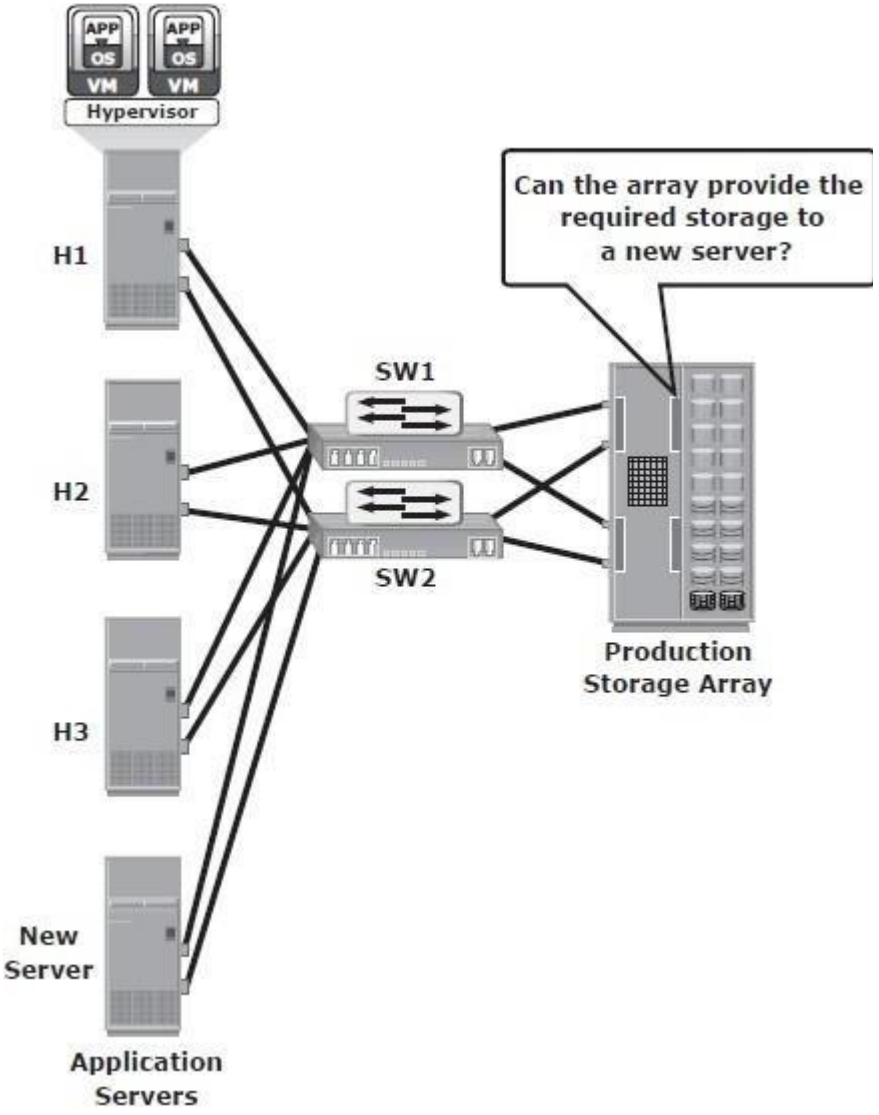


Fig 5.13: Monitoring storage array capacity

- The following example illustrates the importance of monitoring the file system capacity on file servers. Fig 5.14 (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 (Fig 5.14 [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.

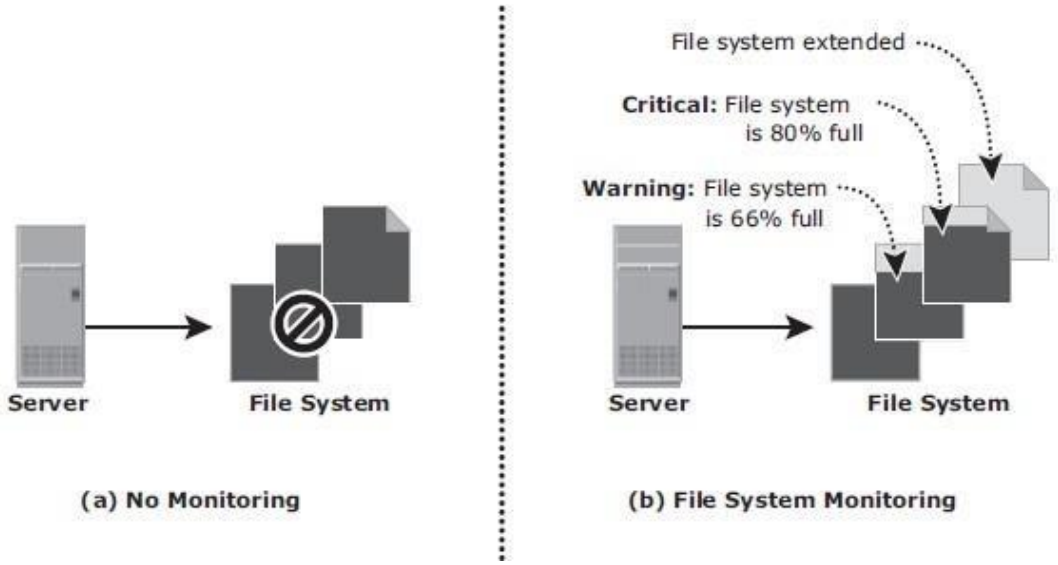


Fig 5.14: Monitoring server file system space

Performance Monitoring

- The example shown in Fig 5.15 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.

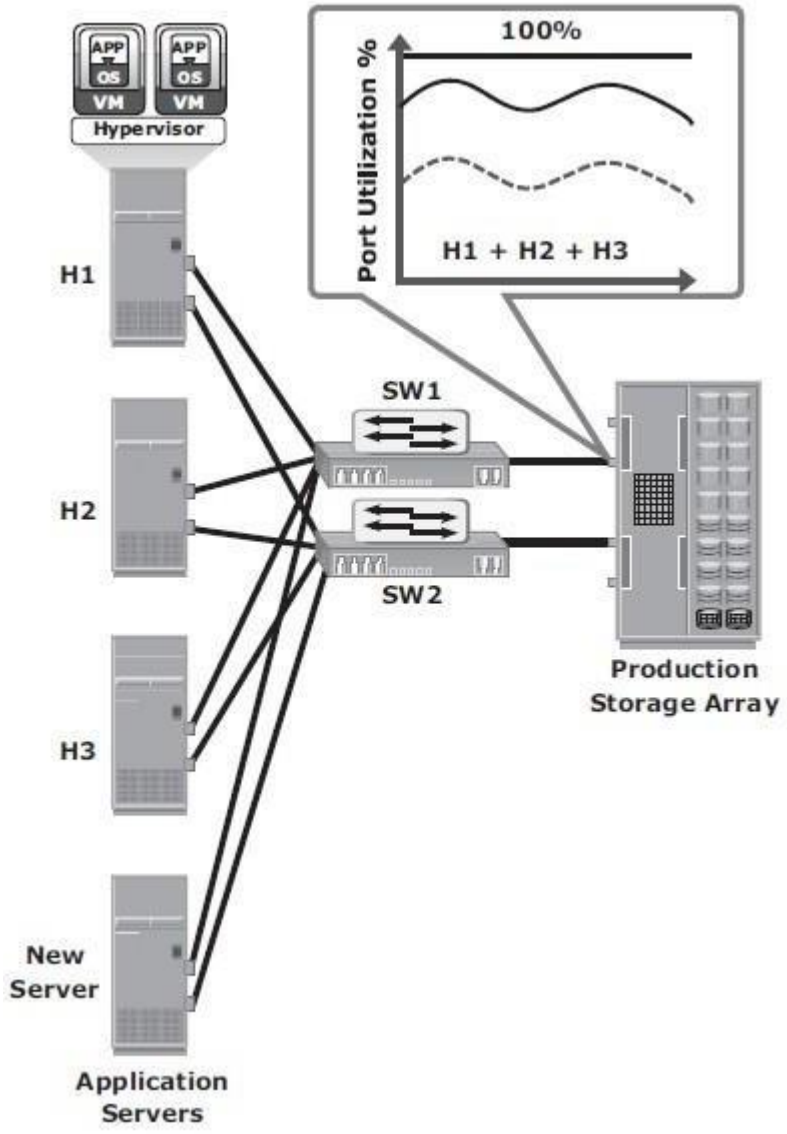


Fig 5.15: Monitoring array port utilization

Security Monitoring

- The example shown in Fig 5.16 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. 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.

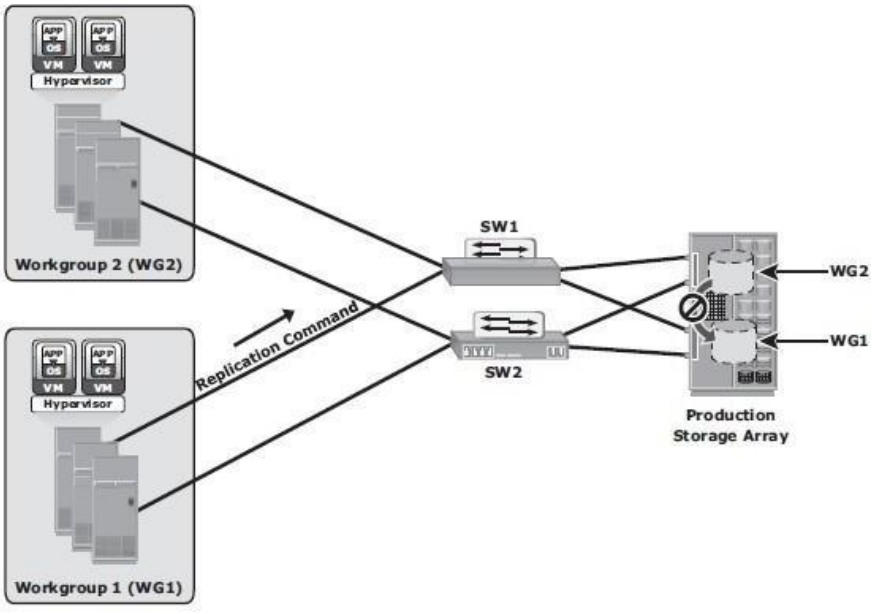


Fig 5.16: Monitoring security in a storage array

5.6.4 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 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.
- **Warning alerts** require administrative attention so that the alerted condition is contained and does not affect accessibility.
- **Fatal alerts** require immediate attention because the condition might affect overall performance, security, or availability.
- 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.

5.7 Storage Infrastructure Management Activities

- 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.

5.7.1 Availability Management

- Availability management requires 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.
- In availability management, the key activity is to provision **redundancy** at all levels, including components, data, or even sites.
- Eg: When a server is deployed to support critical business function, it requires high availability 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.

5.7.2 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 which 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.
- *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.

5.7.3 Performance Management

- **Performance management** ensures the optimal operational efficiency of all components.
- Performance analysis helps to identify the performance of storage infrastructure components and 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.
- For example, to optimize the expected performance levels, *fine-tuning* is required for activities on the server, such as the volume configuration, database design or application layout, configuration of multiple HBAs, and intelligent multipathing software.
- 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.

5.7.4 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.
- The security management task on a storage array includes LUN masking that restricts a host's access to intended LUNs only.

5.7.5 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. It also lists all the equipment, with details of 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.

5.7.6 Storage Infrastructure Management in a Virtualized Environment

- Storage virtualization has enabled dynamic migration of data and extension of storage volumes. Due to dynamic extension, storage volumes can be expanded non-disruptively to meet both capacity and performance requirements.
- Since 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. This results in overutilization or underutilization issues.
- 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 MULTITENANCY

- Multiple tenants sharing the same resources provided by a single landlord (resource provider) is called **multitenancy**.
- Two common examples of multitenancy are:
  - multiple virtual machines sharing the same server hardware through the use of a hypervisor running on the server,
  - multiple user applications using the same storage platform.
- **Security** and **service level assurance** are a key concerns in any multitenant storage environment.
- *Secure multitenancy* means that no tenant can access another tenant's data.
- Below are the four pillars of multitenancy:
  - **Secure separation:** This enables data path separation across various tenants in a multitenant environment. This pillar can be divided into four basic requirements: separation of data at rest, address space separation, authentication and name service separation, and separation of data access.
  - **Service assurance:** Consistent and reliable service levels are integral to storage multitenancy. Service assurance plays an important role in providing service levels that can be unique to each tenant.
  - **Availability:** High availability ensures a resilient architecture that provides fault tolerance and redundancy. This is even more critical when storage infrastructure is shared by multiple tenants, because the impact of any outage is magnified.

- **Management:** This includes provisions that allow a landlord to manage basic infrastructure while delegating management responsibilities to tenants for the resources that they interact with day to day. This concept is known as balancing the provider (landlord) in-control with the tenant in-control capabilities.

5.7.7 Storage Management Examples

Example 1: Storage Allocation to a New Server/Host

- Consider the deployment of a new RDBMS server to the existing **nonvirtualized storage infrastructure environment**.
- Below are the storage management activities, performed by the administrator:

1. Install and configure the HBAs and device drivers on the server before it is physically connected to the SAN. Multipathing software can also be installed on the server.
2. Connect storage array ports to the SAN and perform zoning on the SAN switches to allow the new server access to the storage array ports via its HBAs.
3. Ensure redundant paths between the server and the storage array by connecting the HBAs of the new server to different switches and zoning with different array ports.
4. Configure LUNs on the array and assign these LUNs to the storage array front- end ports. LUN masking configuration is performed on the storage array, which restricts access to LUNs by a specific server.
5. 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.
6. 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.
7. Install database or an application on the logical volumes or file systems that were created.
8. The last step is to make the database or application capable of using the new file system space.

- Fig 5.17 illustrates the activities performed on a server, a SAN, and a storage array for the allocation of storage to a new server.

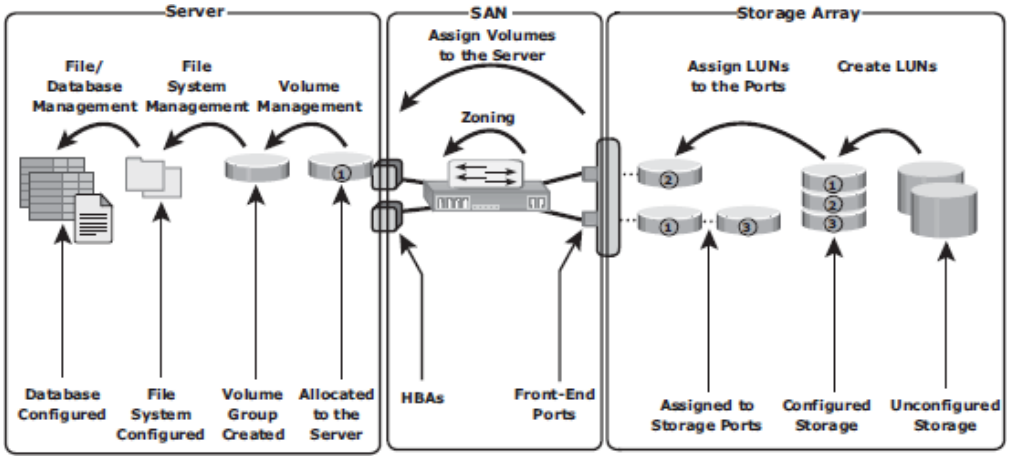


Fig 5.17: Storage allocation tasks

- Below are the various administrative tasks performed in a **virtualized environment** to provision storage to a VM that runs an RDBMS.

1. Similar to nonvirtualized environment, a physical connection must be established between the physical server, which hosts the VMs, and the storage array through the SAN.
2. At the SAN level, a VSAN can be configured to transfer data between the physical server and the storage array. This isolates storage traffic from any other traffic in the SAN. Zoning can be configured within the VSAN.
3. 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. LUN masking needs to be carried out on the storage array.
4. 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.
5. Administrator creates a VM and installs the OS and RDBMS on the VM. During this, 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.
6. 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.
- Fig 5.18 shows the steps and considerations for the extension of file systems in the flow chart.

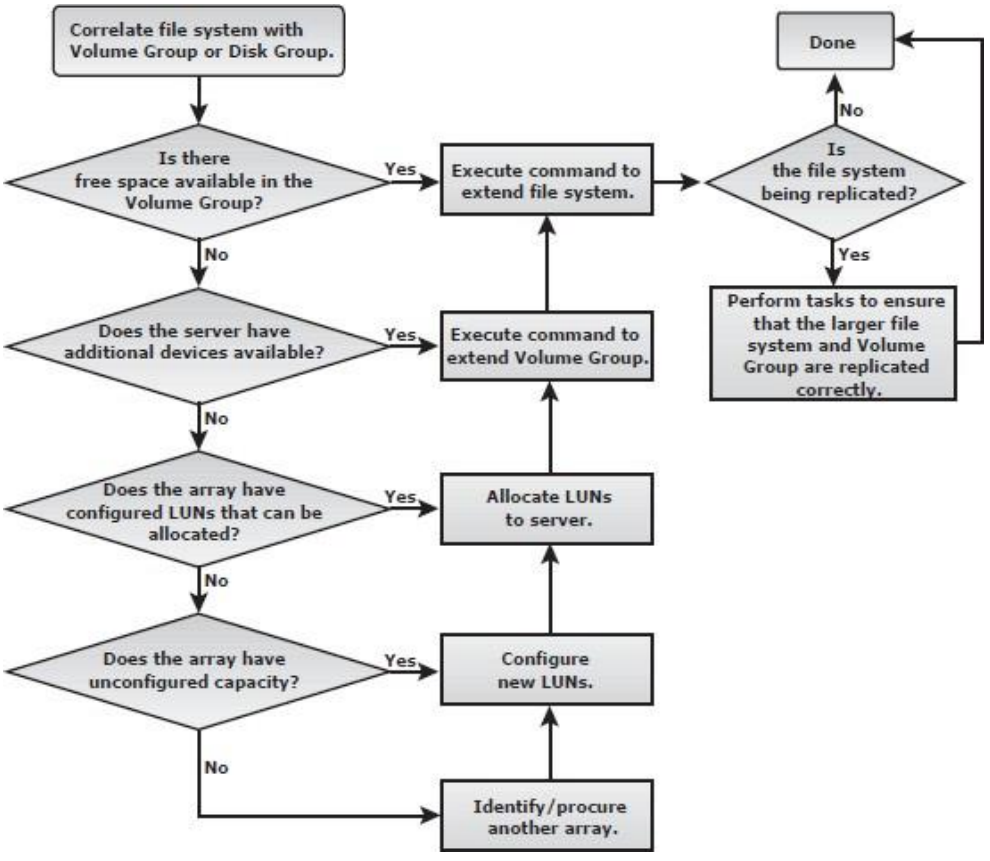


Fig 5.18: Extending a file system

Example 3: Chargeback Report

- This example explores the storage infrastructure management tasks necessary to create a **chargeback report**.
- Fig 5.19 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.
- 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.
- Individual departmental applications run on each of the servers.
- 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.
- Fig 5.19 shows a sample report for two applications, *Payroll_1* and *Engineering_1*.
- The first step to determine chargeback costs is to correlate the application with the exact amount of raw storage configured for that application.
- Fig 5.20 shows the storage space used for Payroll_1 application identified based on 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, Payroll_1 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).
- Based on 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.
- The **raw storage** configured for the application is determined from the allocated storage based on the RAID protection that is used.

- If the Payroll_1 application's production volumes are RAID 1-protected, the raw space used is 200 GB.

  Assume 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.

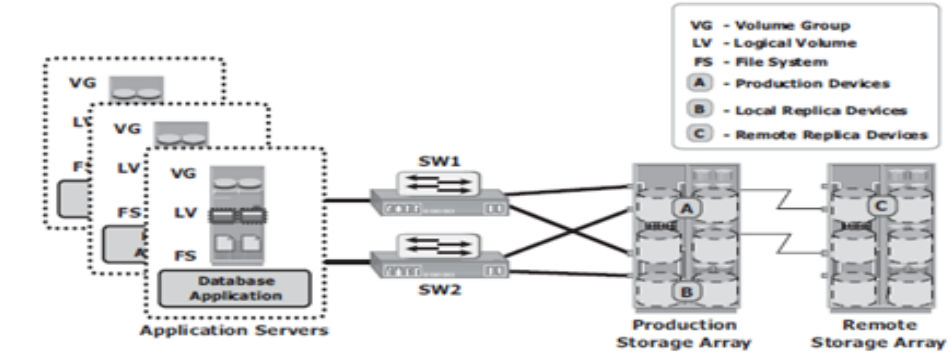
- Therefore, the total raw capacity used by the Payroll_1 application is 425 GB.

  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 (eg: Engineering_1, etc) 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



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

Fig 5.19: Configuration and Chargeback report

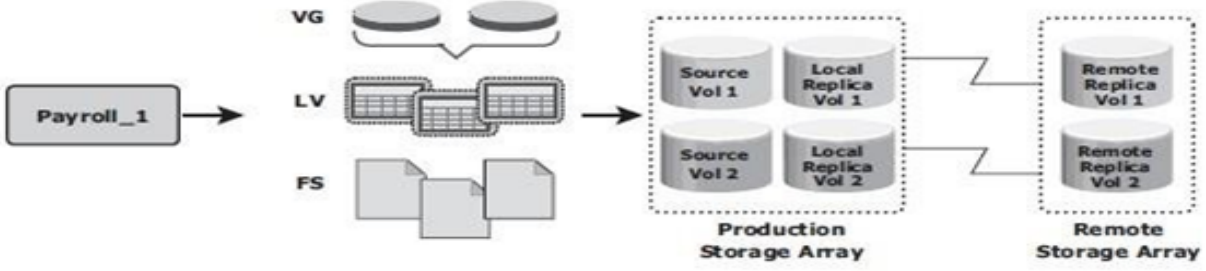


Fig 5.20: Correlation of capacity configured for an application

5.8 Storage Infrastructure Management Challenges

- The main challenge in monitoring and managing today's complex storage infrastructure is due to the heterogeneity of storage arrays, networks, servers, databases, and applications in the environment.
- Eg: 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.

5.9 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 information 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 Fig 5.21).
- 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.

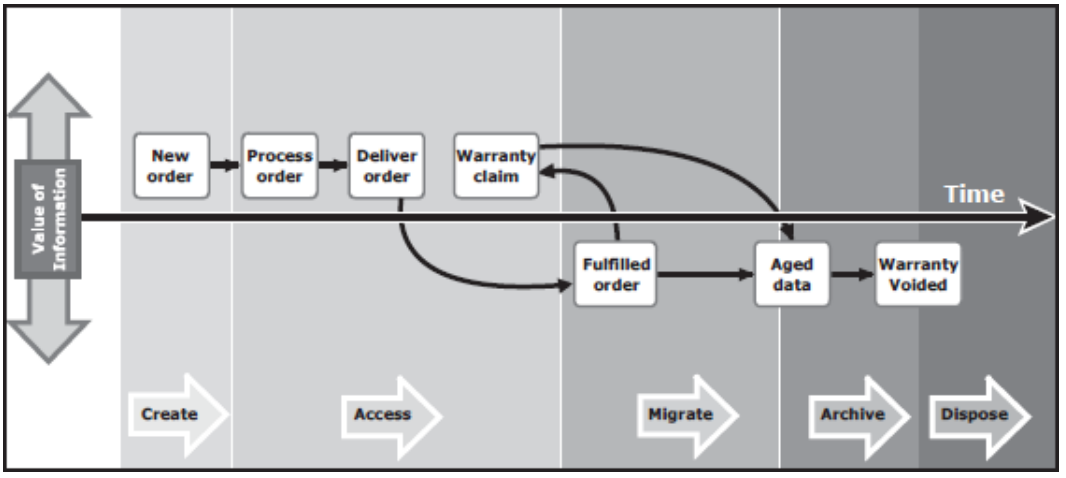


Fig 5.21 Changing value of sales order information

- **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

5.10 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 solidstate 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.

5.10.1 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 explained next.
- Traditionally, storage tiering is operated at the LUN level that moves an entire LUN from one tier of storage to another (see Fig 5.22 [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 Fig 5.22 [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.

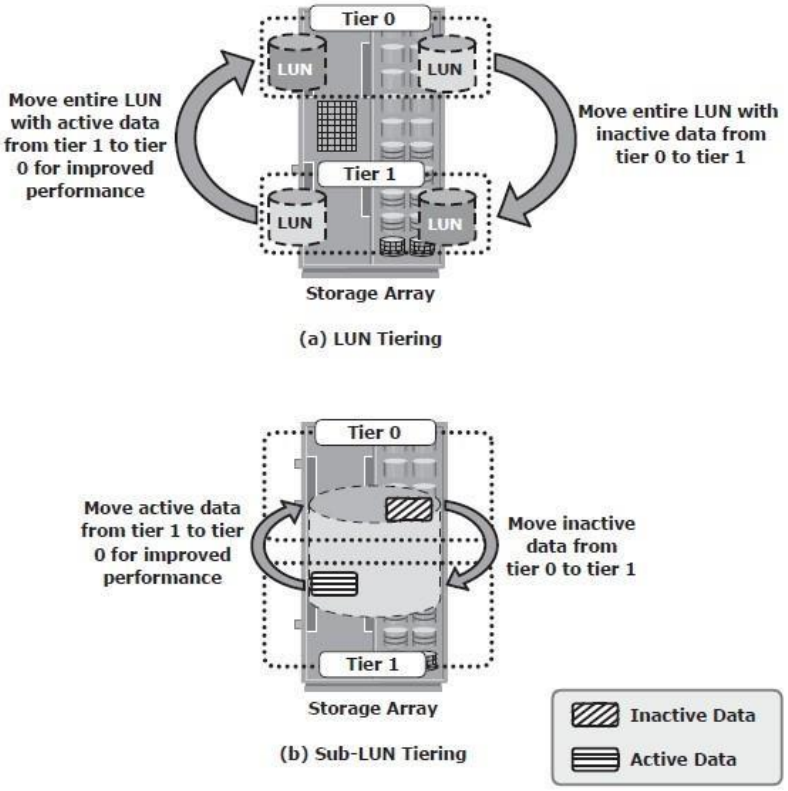


Fig 5.22: Implementation of intra-array storage tiering

15.10.2 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 5.23 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.

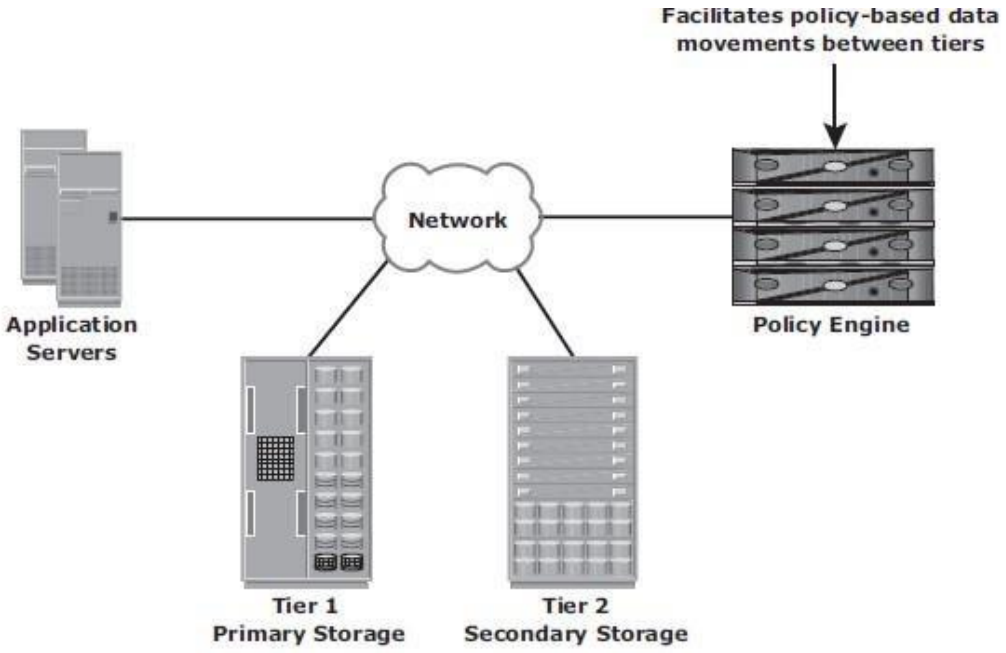


Fig 5.23: Implementation of intra-array storage tiering