MODULE – 3

BACKUP, ARCHIVE, AND REPLICATION

3.1 INTRODUCTION TO BUSINESS CONTINUITY

Business Continuity (BC):

Business continuity (BC) is an integrated and enterprise wide process that includes all activities (internal and external to IT) that a business must perform to mitigate the impact of planned and unplanned downtime.

BC entails preparing for, responding to, and recovering from a system outage that adversely affects business operations. It involves proactive measures, such as business impact analysis, risk assessments, deployment of BC technology solutions (backup and replication), and reactive measures, such as disaster recovery and restart, to be invoked in the event of a failure.

The goal of a BC solution is to ensure the **"information availability"** required to conduct vital business operations.

3.1.1 Information Availability:

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

1. Reliability,
2. Accessibility
3. Timeliness.

1. **Reliability:** This reflects a component's ability to function without failure, under stated conditions, for a specified amount of time.
2. **Accessibility:** This is the state within which the required information is accessible at the right place, to the right user. The period of time during which the system is in an accessible state is termed **system uptime;** when it is not accessible it is termed **system**

downtime.

3. **Timeliness:** Defines the exact moment or the time window (a particular time of the day, week, month, and/or year as specified) during which information must be accessible. For example, if online access to an application is required between 8:00 am and 10:00 pm each day, any disruptions to data availability outside of this time slot are not considered to affect timeliness.

3.1.1.1 Causes of Information Unavailability

Various planned and unplanned incidents result in data unavailability.

- **Planned outages** include installation/integration/maintenance of new hardware, software upgrades or patches, taking backups, application and data restores, facility operations (renovation and construction), and refresh/migration of the testing to the production environment.
- **Unplanned outages** include failure caused by database corruption, component failure, and human errors.
- **Disasters (natural or man-made)** such as flood, fire, earthquake, and contamination are another type of incident that may cause data unavailability.

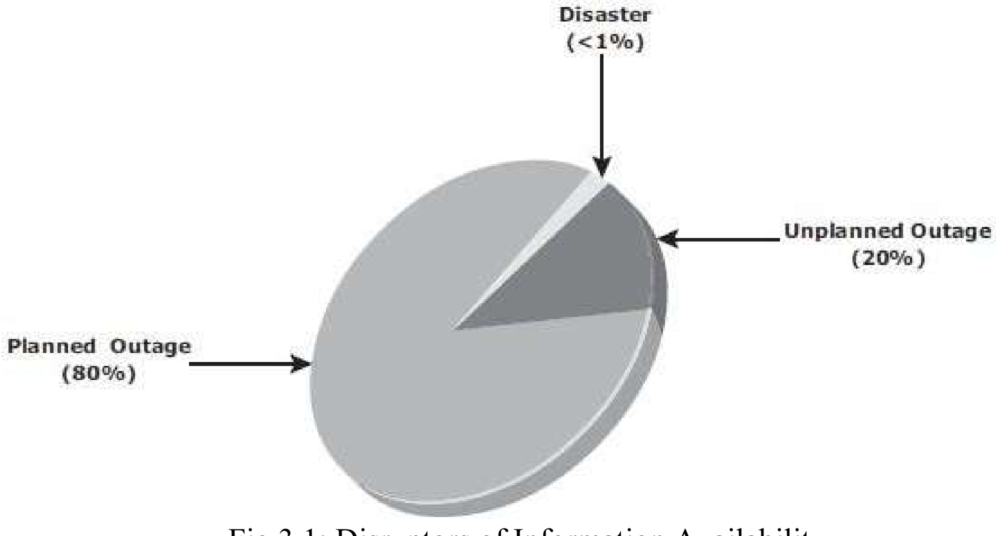


Fig 3.1: Disruptors of Information Availability

As illustrated in Fig 3.1 above, the majority of outages are planned. Planned outages are expected and scheduled, but still cause data to be unavailable.

3.1.1.2 Consequences of Downtime

- Information unavailability or downtime results in loss of productivity, loss of revenue, poor financial performance, and damage to reputation.
- Loss of productivity includes reduced output per unit of labor, equipment, and capital.
- Loss of revenue includes direct loss, compensatory payments, future revenue loss, billing loss, and investment loss.
- Poor financial performance affects revenue recognition, cash flow, discounts, payment guarantees, credit rating, and stock price.
- Damages to reputations may result in a loss of confidence or credibility with customers, suppliers, financial markets, banks, and business partners.
- An important metric, *average cost of downtime per hour*, provides a key estimate in determining the appropriate BC solutions. It is calculated as follows:

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

  Where:

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

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

3.1.1.3 Measuring Information Availability

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

→ **Mean Time Between Failure (MTBF):** It is the average time available for a system or component to perform its normal operations between failures.

→ **Mean Time To Repair (MTTR):** It is the average time required to repair a failed component. MTTR includes the total time required to do the following activities: Detect the fault, mobilize the maintenance team, diagnose the fault, obtain the spare parts, repair, test, and restore the data.

Fig 3.2 illustrates the various information availability metrics that represent system uptime and downtime.

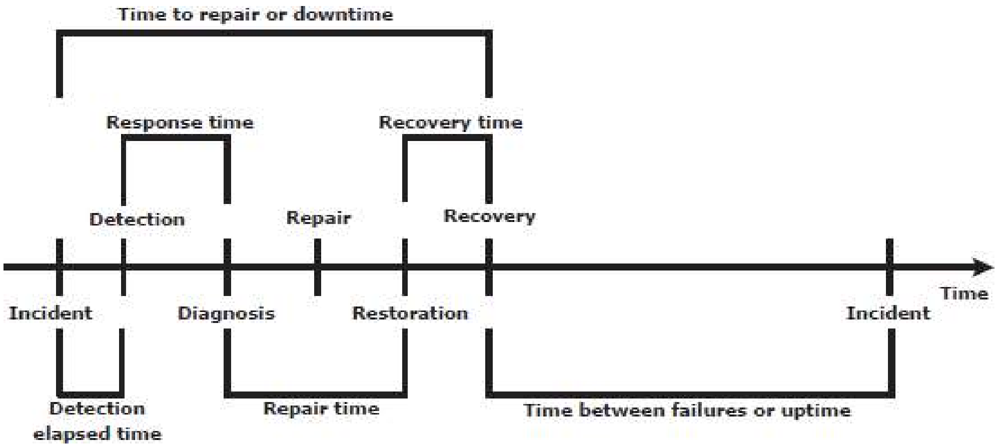


Fig 3-2: Information availability metrics

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

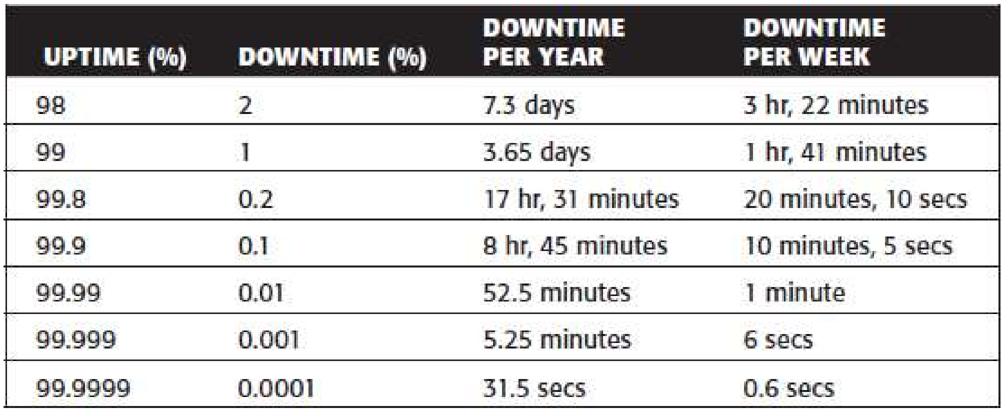
IA = system uptime / (system uptime + system downtime)

In terms of MTBF and MTTR, IA could also be expressed as

IA = MTBF / (MTBF + MTTR)

Uptime per year is based on the exact timeliness requirements of the service, this calculation leads to the number of “9s” representation for availability metrics.

Table 3-1 lists the approximate amount of downtime allowed for a service to achieve certain levels of 9s availability. For example, a service that is said to be “five 9s available” is available for 99.999 percent of the scheduled time in a year (24 × 365).



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

Table 3-1: Availability percentage and Allowable downtime

3.1.2 BC Terminology

This section defines common terms related to BC operations which are used in this module to explain advanced concepts:

- **Disaster recovery:** This is the coordinated process of restoring systems, data, and the infrastructure required to support key ongoing business operations in the event of a disaster. It is the process of restoring a previous copy of the data and applying logs or other necessary processes to that copy to bring it to a known point of consistency. Once all recoveries are completed, the data is validated to ensure that it is correct.
- **Disaster restart:** This is the process of restarting business operations with mirrored consistent copies of data and applications.
- **Recovery-Point Objective (RPO):** This is the point in time to which systems and data must be recovered after an outage. It defines the amount of data loss that a business can endure. A large RPO signifies high tolerance to information loss in a business. Based on the RPO, organizations plan for the minimum frequency with which a backup or replica must be made. For example, if the RPO is six hours, backups or replicas must be made at least once in 6 hours. Fig 3.3 (a) shows various RPOs and their corresponding ideal recovery strategies. An organization can plan for an appropriate BC technology solution on the basis of the RPO it sets. For example:

  → **RPO of 24 hours:** This ensures that backups are created on an offsite tape drive every midnight. The corresponding recovery strategy is to restore data from the set of last backup tapes.

→ **RPO of 1 hour:** Shipping database logs to the remote site every hour. The corresponding recovery strategy is to recover the database at the point of the last log shipment.

→ **RPO in the order of minutes:** Mirroring data asynchronously to a remote site

→ **Near zero RPO:** This mirrors mission-critical data synchronously to a remote site.

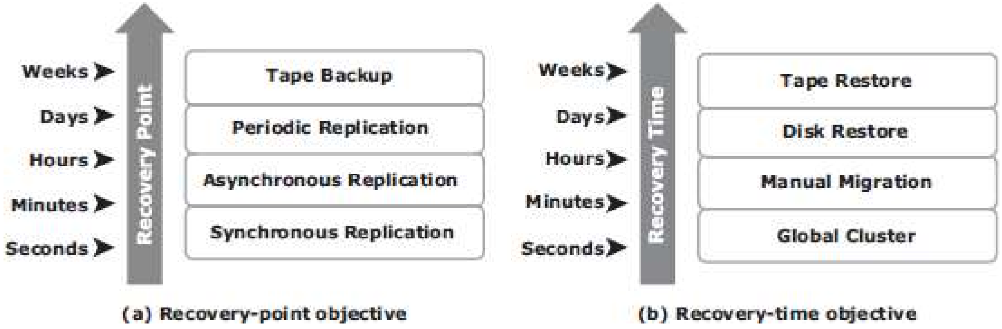


Fig 3.3: Strategies to meet RPO and RTO targets

- **Recovery-Time Objective (RTO):** The time within which systems and applications must be recovered after an outage. It defines the amount of downtime that a business can endure and survive. Businesses can optimize disaster recovery plans after defining the RTO for a given system. For example, if the RTO is two hours, then use a disk backup because it enables a faster restore than a tape backup. However, for an RTO of one week, tape backup will likely meet requirements. Some examples of RTOs and the recovery strategies to ensure data availability are listed below (refer to Fig 3.3 (b)):

→ **RTO of 72 hours:** Restore from backup tapes at a cold site.

→ **RTO of 12 hours:** Restore from tapes at a hot site.

→ **RTO of few hours:** Use a data vault to a hot site.

→ **RTO of a few seconds:** Cluster production servers with bidirectional mirroring, enabling the applications to run at both sites simultaneously.

3.1.3 BC Planning Life Cycle

BC planning must follow a disciplined approach like any other planning process. Organizations today dedicate specialized resources to develop and maintain BC plans. From the conceptualization to the realization of the BC plan, a life cycle of activities can be defined for the BC process.

The BC planning lifecycle includes five stages shown below (Fig 3.4):

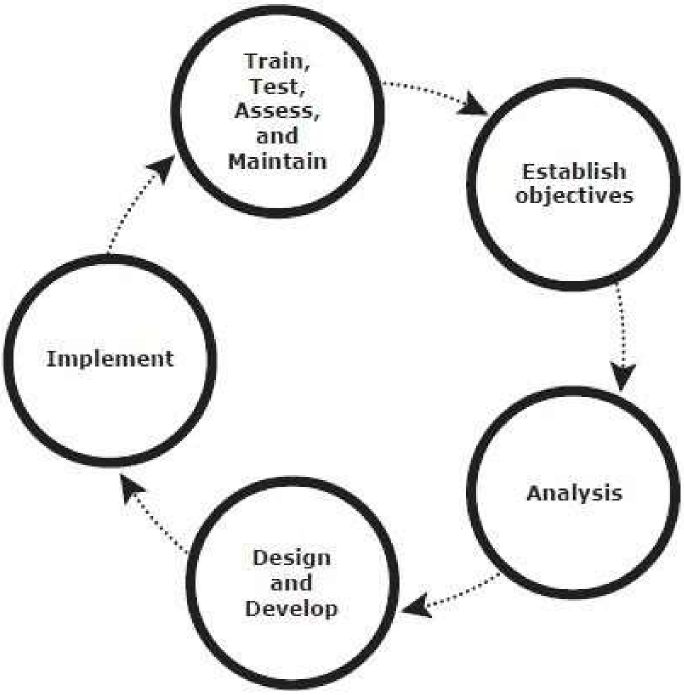


Fig 3.4: BC Planning Lifecycle

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

1. **Establishing objectives**

→ Determine BC requirements.

→ Estimate the scope and budget to achieve requirements.

→ Select a BC team by considering subject matter experts from all areas of the business, whether internal or external.

→ Create BC policies.

2. Analyzing

→ Collect information on data profiles, business processes, infrastructure support, dependencies, and frequency of using business infrastructure.

→ Identify critical business needs and assign recovery priorities.

→ Create a risk analysis for critical areas and mitigation strategies.

→ Conduct a Business Impact Analysis (BIA).

→ Create a cost and benefit analysis based on the consequences of data unavailability.

3. Designing and developing

→ Define the team structure and assign individual roles and responsibilities. For example, different teams are formed for activities such as emergency response, damage assessment, and infrastructure and application recovery.

→ Design data protection strategies and develop infrastructure.

→ Develop contingency scenarios.

→ Develop emergency response procedures.

→ Detail recovery and restart procedures.

4. Implementing

→ Implement risk management and mitigation procedures that include backup, replication, and management of resources.

→ Prepare the disaster recovery sites that can be utilized if a disaster affects the primary data center.

→ Implement redundancy for every resource in a data center to avoid single points of failure.

5. Training, testing, assessing, and maintaining

→ Train the employees who are responsible for backup and replication of business-critical data on a regular basis or whenever there is a modification in the BC plan

→ Train employees on emergency response procedures when disasters are declared.

→ Train the recovery team on recovery procedures based on contingency scenarios.

→ Perform damage assessment processes and review recovery plans.

→ Test the BC plan regularly to evaluate its performance and identify its limitations.

→ Assess the performance reports and identify limitations.

→ Update the BC plans and recovery/restart procedures to reflect regular changes within the data center.

3.1.4 Failure Analysis

3.1.4.1 Single Point of Failure

- A **single point of failure** refers to the failure of a component that can terminate the availability of the entire system or IT service.
- Fig 3.5 depicts a system setup in which an application, running on a VM, provides an interface to the client and performs I/O operations.
- The client is connected to the server through an IP network, the server is connected to the storage array through a FC connection, an HBA installed at the server sends or receives data to and from a storage array, and an FC switch connects the HBA to the storage port
- In a setup where **each component must function as required to ensure data availability**, the failure of a single physical or virtual component causes the failure of the entire data center or an application, resulting in disruption of business operations.
- In this example, failure of a hypervisor can affect all the running VMs and the virtual network, which are hosted on it.
- The can be several similar single points of failure identified in this example. A VM, a hypervisor, an HBA/NIC on the server, the physical server, the IP network, the FC switch, the storage array ports, or even the storage array could be a potential single point of failure. To avoid single points of failure, it is essential to implement a fault-tolerant mechanism.

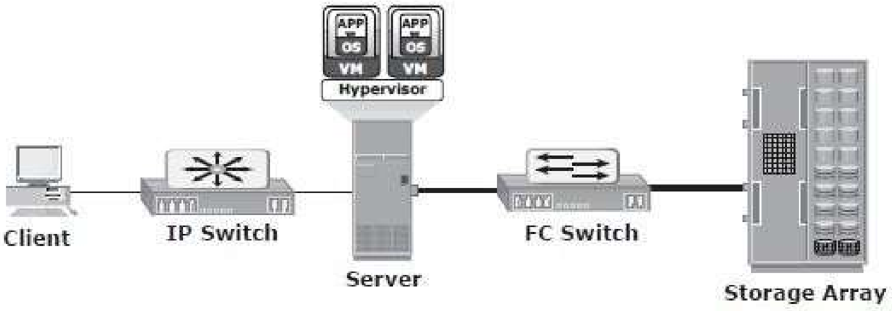


Fig 3.5: Single Point of Failure

3.1.4.2 Resolving Single Points of Failure

- To mitigate a single point of failure, systems are designed with redundancy, such that the system will fail only if all the components in the redundancy group fail. This ensures that the failure of a single component does not affect data availability.

- Data centers follow stringent guidelines to implement fault tolerance for uninterrupted information availability. Careful analysis is performed to eliminate every single point of failure.
- The example shown in Fig 3.6 represents all enhancements of the system shown in Fig 3.5 in the infrastructure to mitigate single points of failure:
    - Configuration of redundant HBAs at a server to mitigate single HBA failure
    - Configuration of NIC (network interface card) teaming at a server allows protection against single physical NIC failure. It allows grouping of two or more physical NICs and treating them as a single logical device. NIC teaming eliminates the single point of failure associated with a single physical NIC.
    - Configuration of redundant switches to account for a switch failure
    - Configuration of multiple storage array ports to mitigate a port failure
    - RAID and hot spare configuration to ensure continuous operation in the event of disk failure
    - Implementation of a redundant storage array at a remote site to mitigate local site failure
    - Implementing server (or compute) clustering, a fault-tolerance mechanism whereby two or more servers in a cluster access the same set of data volumes. Clustered servers exchange a heartbeat to inform each other about their health. If one of the servers or hypervisors fails, the other server or hypervisor can take up the workload.
    - Implementing a VM Fault Tolerance mechanism ensures BC in the event of a server failure. This technique creates duplicate copies of each VM on another server so that when a VM failure is detected, the duplicate VM can be used for failover. The two VMs are kept in synchronization with each other in order to perform successful failover.

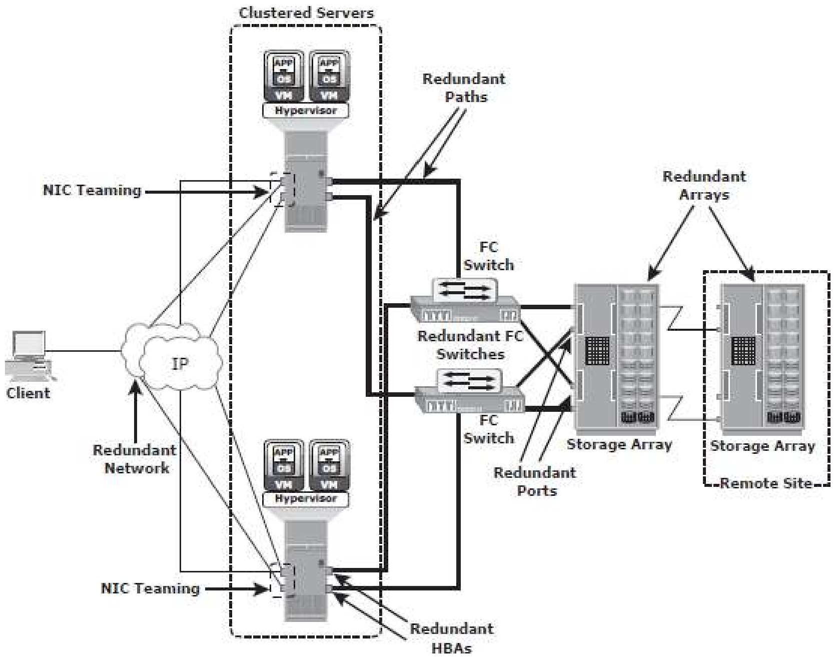


Fig 3.6: Resolving single points of failure

3.1.4.3 Multipathing Software

- Configuration of multiple paths increases the data availability through path failover. If servers are configured with one I/O path to the data there will be no access to the data if that path fails. Redundant paths eliminate the path to become single points of failure.
- Multiple paths to data also improve I/O performance through load sharing and maximize server, storage, and data path utilization.
- In practice, merely configuring multiple paths does not serve the purpose. Even with multiple paths, if one path fails, I/O will not reroute unless the system recognizes that it has an alternate path.
- Multipathing software provides the functionality to recognize and utilize alternate I/O path to data. Multipathing software also manages the load balancing by distributing I/Os to all available, active paths.
- In a virtual environment, multipathing is enabled either by using the hypervisor's built-in capability or by running a third-party software module, added to the hypervisor.

3.1.5 BC Technology Solutions

After analyzing the business impact of an outage, designing appropriate solutions to recover from a failure is the next important activity. One or more copies of the original data are maintained using any of the following strategies, so that data can be recovered and business operations can be restarted using an alternate copy:

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

3.2 Backup and Recovery

- **Data Backup** is a copy of production data, created and retained for the sole purpose of recovering lost or corrupted data.
- Evaluating the various backup methods along with their recovery considerations and retention requirements is an essential step to implement a successful backup and recovery solution.
- Organizations generate and maintain large volumes of data, and most of the data is fixed content. This fixed content is rarely accessed after a period of time. Still, this data needs to be retained for several years to meet regulatory compliance.
- **Data archiving** is the process of moving data that is no longer actively used, from primary storage to a low-cost secondary storage. This data is retained in the secondary storage for a long term to meet regulatory requirements. This reduces the amount of data to be backed up and the time required to back up the data.

3.2.1 Backup Purpose

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

3.2.1.1 Disaster Recovery

- Backups are performed to address disaster recovery needs.
- The backup copies are used for restoring data at an alternate site when the primary site is incapacitated due to a disaster. Based on RPO and RTO requirements, organizations use different backup strategies for disaster recovery.
- When a tape-based backup method is used as a disaster recovery strategy, the backup tape media is shipped and stored at an offsite location. These tapes can be recalled for restoration at the disaster recovery site.
- Organizations with stringent RPO and RTO requirements use remote replication technology to replicate data to a disaster recovery site. Organizations can bring production systems online in a relatively short period of time if a disaster occurs.

3.2.1.2 Operational Recovery

- Data in the production environment changes with every business transaction and operation.
- Operational recovery is the use of backups to restore data if data loss or logical

corruption occurs during routine processing.

- For example, it is common for a user to accidentally delete an important email or for a file to become corrupted, which can be restored from operational backup.

3.2.1.3 Archival

- Backups are also performed to address archival requirements.
- Traditional backups are still used by small and medium enterprises for long-term preservation of transaction records, e- mail messages, and other business records required for regulatory compliance.

Apart from addressing disaster recovery, archival, and operational requirements, backups serve as a protection against data loss due to physical damage of a storage device, software failures, or virus attacks. Backups can also be used to protect against accidents such as a deletion or intentional data destruction.

3.2.2 Backup Methods

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

- Consistent backups of databases can also be done by using a cold backup. This requires the database to remain inactive during the backup. Of course, the disadvantage of a cold backup is that the database is inaccessible to users during the backup process.
- Hot backup is used in situations where it is not possible to shut down the database. This is facilitated by database backup agents that can perform a backup while the database is active. The disadvantage associated with a hot backup is that the agents usually affect overall application performance.
- A **point-in-time (PIT)** copy method is deployed in environments where the impact of downtime from a cold backup or the performance resulting from a hot backup is unacceptable. The PIT copy is created from the production volume and used as the source for the backup. This reduces the impact on the production volume.
- Certain attributes and properties attached to a file, such as permissions, owner, and other metadata, also need to be backed up. These attributes are as important as the data itself and must be backed up for consistency.
- Backup of boot sector and partition layout information is also critical for successful recovery.
- In a disaster recovery environment, **bare-metal recovery (BMR)** refers to a backup in which all metadata, system information, and application configurations are appropriately backed up for a full system recovery. BMR builds the base system, which includes partitioning, the file system layout, the operating system, the applications, and all the relevant configurations. BMR recovers the base system first, before starting the recovery of data files. Some BMR technologies can recover a server onto dissimilar hardware.

3.2.3 Backup Topologies

- Three basic topologies are used in a backup environment:

1. Direct attached backup
2. LAN based backup, and
3. SAN based backup.

- A **mixed topology** is also used by combining LAN based and SAN based topologies.
- In a **direct-attached backup**, a backup device is attached directly to the client. Only the metadata is sent to the backup server through the LAN. This configuration frees the LAN from backup traffic.
- The example shown in Fig 3.7 device is directly attached and dedicated to the backup client. As the environment grows, however, there will be a need for central management of all backup devices and to share the resources to optimize costs. An appropriate solution is to share the backup devices among multiple servers. Network-based topologies (LAN-based and SAN-based) provide the solution to optimize the utilization of backup devices.

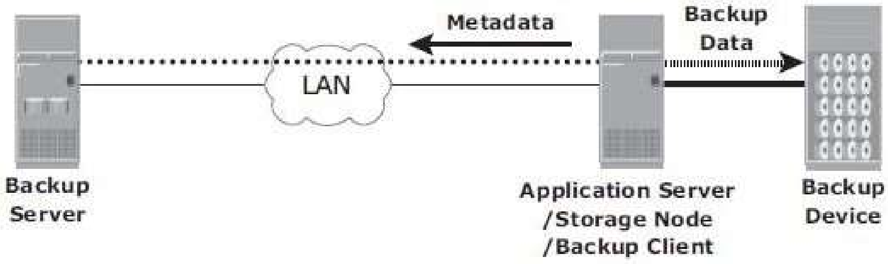


Fig 3.7: Direct-attached backup topology

- In **LAN-based backup**, the clients, backup server, storage node, and backup device are connected to the LAN (see Fig 3.8). The data to be backed up is transferred from the backup client (source), to the backup device (destination) over the LAN, which may affect network performance.
- This impact can be minimized by adopting a number of measures, such as configuring separate networks for backup and installing dedicated storage nodes for some application servers.

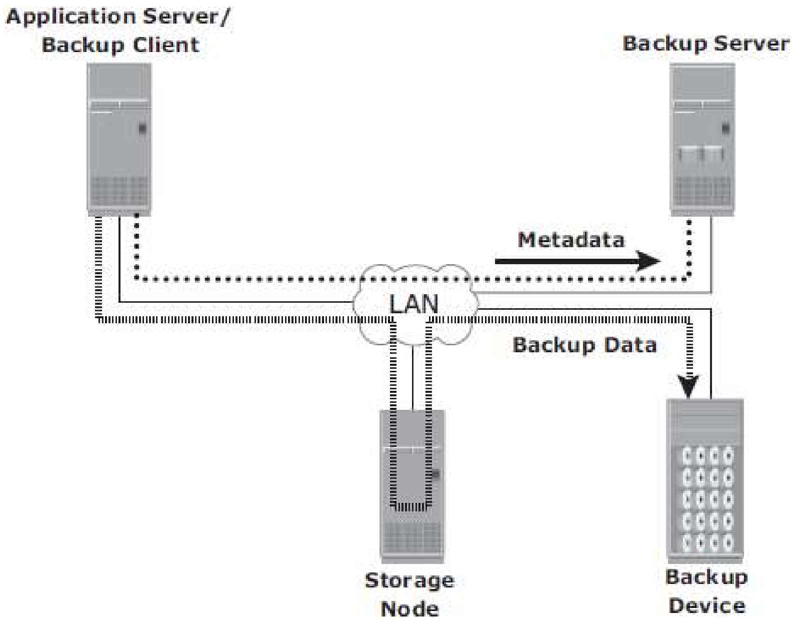


Fig 3.8: LAN-based backup topology

- The **SAN-based backup** is also known as the *LAN-free backup*. Fig 3.9 illustrates a SAN-based backup. The SAN-based backup topology is the most appropriate solution when a backup device needs to be shared among the clients. In this case the backup device and clients are attached to the SAN.
- In the example from Fig 3.9, a client sends the data to be backed up to the backup device over the SAN. Therefore, the backup data traffic is restricted to the SAN, and only the backup metadata is transported over the LAN. The volume of metadata is insignificant when compared to the production data; the LAN performance is not degraded in this configuration.

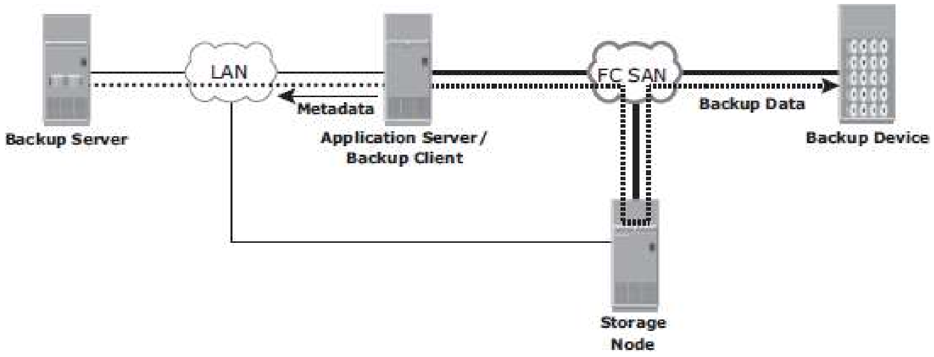


Fig 3.9: SAN-based backup topology

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

- The mixed topology uses both the LAN-based and SAN-based topologies, as shown in Fig 3.10. This topology might be implemented for several reasons, including cost, server location, reduction in administrative overhead, and performance considerations.

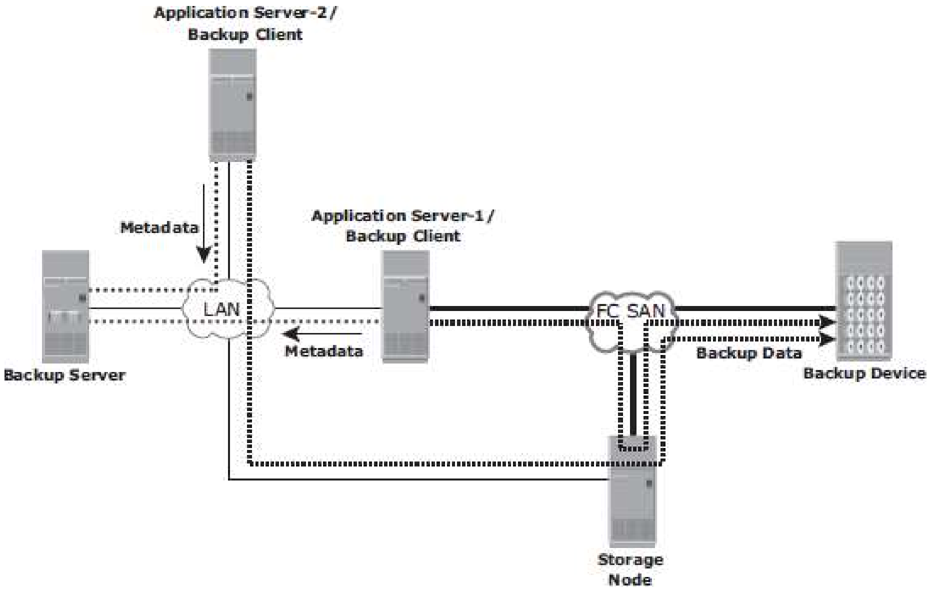


Fig 3.10: Mixed backup topology

3.2.4 Backup Technologies

- A wide range of technology solutions are currently available for backup targets.
- Tapes and disks are the two most commonly used backup media. Virtual tape libraries use disks as backup medium emulating tapes, providing enhanced backup and recovery capabilities.

3.2.4.1 Backup to Tape

- Tapes, a low-cost technology, are used extensively for backup. Tape drives are used to read/write data from/to a tape cartridge. Tape drives are referred to as sequential, or linear, access devices because the data is written or read sequentially.
- A tape cartridge is composed of magnetic tapes in a plastic enclosure.
- Tape Mounting is the process of inserting a tape cartridge into a tape drive. The tape drive has motorized controls to move the magnetic tape around, enabling the head to read or write data.
- Several types of tape cartridges are available. They vary in size, capacity, shape, number of reels, density, tape length, tape thickness, tape tracks, and supportedspeed.

Physical Tape Library

- The physical tape library provides housing and power for a number of tape drives and tape cartridges, along with a robotic arm or picker mechanism.
- The backup software has intelligence to manage the robotic arm and entire backup process. Fig 3-14 shows a physical tape library.
- *Tape drives* read and write data from and to a tape. Tape cartridges are placed in the slots when not in use by a tape drive. *Robotic arms* are used to move tapes around the library, such as moving a tape drive into a slot.

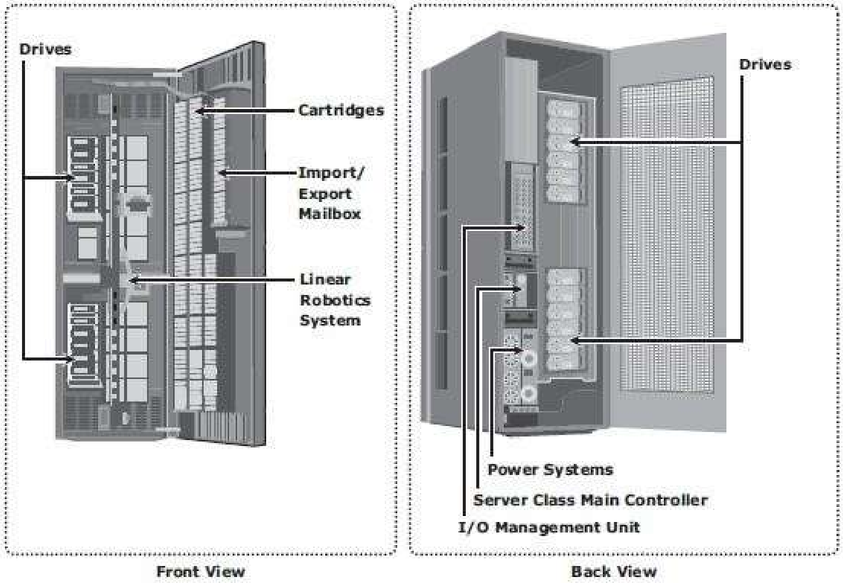


Fig 3.11: Physical tape library

- Another type of slot called a *mail or import/export* slot is used to add or remove tapes from the library without opening the access doors (Fig 3.11 Front View) because opening the access doors causes a library to go offline.
- In addition, each physical component in a tape library has an individual element address that is used as an addressing mechanism for moving tapes around the library.
- When a backup process starts, the robotic arm is instructed to load a tape to a tape drive. This process adds to the delay to a degree depending on the type of hardware used, but it generally takes 5 to 10 seconds to mount a tape. After the tape is mounted, additional time is spent to position the heads and validate header information. This total time is called *load to ready time*, and it can vary from several seconds to minutes.
- The tape drive receives backup data and stores the data in its internal buffer. This backup data is then written to the tape in blocks. During this process, it is best to ensure that the tape drive is kept busy continuously to prevent gaps between the blocks. This is

accomplished by buffering the data on tape drives.

- The speed of the tape drives can also be adjusted to match data transfer rates.
- Tape drive *streaming or multiple streaming* writes data from multiple streams on a single tape to keep the drive busy. Shown in Fig 3.12, multiple streaming improves media performance, but it has an associated disadvantage. The backup data is interleaved because data from multiple streams is written on it.

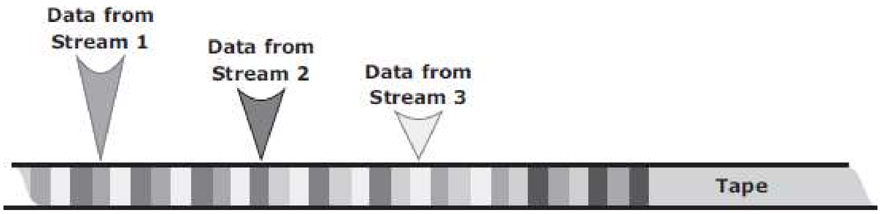


Fig 3.12: Physical tape library

- Many times, even the buffering and speed adjustment features of a tape drive fail to prevent the gaps, causing the *“shoe shining effect” or “backhitching.”* This is the repeated back and forth motion a tape drive makes when there is an interruption in the backup data stream. This repeated back-and-forth motion not only causes a degradation of service, but also excessive wear and tear to tapes.
- When the tape operation finishes, the tape rewinds to the starting position and it is unmounted. The robotic arm is then instructed to move the unmounted tape back to the slot. *Rewind time* can range from several seconds to minutes.
- When a *restore* is initiated, the backup software identifies which tapes are required. The robotic arm is instructed to move the tape from its slot to a tape drive. If the required tape is not found in the tape library, the backup software displays a message, instructing the operator to manually insert the required tape in the tape library.
- When a file or a group of files require restores, the tape must move sequentially to the beginning of the data before it can start reading. This process can take a significant amount of time, especially if the required files are recorded at the end of the tape.
- Modern tape devices have an indexing mechanism that enables a tape to be fast forwarded to a location near the required data.

Limitations of Tape

- Tapes must be stored in locations with a controlled environment to ensure preservation of the media and prevent data corruption.
- Data access in a tape is sequential, which can slow backup and recovery operations.
- Physical transportation of the tapes to offsite locations also adds management overhead.

3.2.4.2 Backup to Disk

- Because of *increased availability*, low cost **disks** have now replaced tapes as the primary device for storing backup data because of their *performance advantages*. Backup-to-disk systems offer *ease of implementation*, *reduced TCO* (Total cost of ownership), and *improved quality of service*. Disks also offer *faster recovery* when compared to tapes.
- Backing up to disk storage systems offers clear advantages due to their inherent random access and RAID-protection capabilities.
- Fig 3.13 illustrates a recovery scenario comparing tape versus disk in a Microsoft Exchange environment that supports 800 users with a 75 MB mailbox size and a 60 GB database. As shown, a restore from disk took 24 minutes compared to the restore from a tape, which took 108 minutes for the same environment.
- Recovering from a full backup copy stored on disk and kept onsite provides the fastest recovery solution. Using a disk enables the creation of full backups more frequently, which in turn improves RPO and RTO.
- Backup to disk does not offer any inherent offsite capability, and is dependent on other technologies such as local and remote replication.
- Some backup products also require additional modules and licenses to support backup to disk, which may also require additional configuration steps, including creation of RAID groups and file system tuning. These activities are not usually performed by a backup administrator.

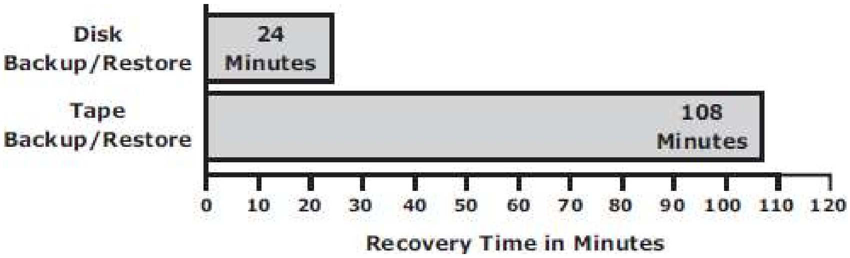


Fig 3.13: Tape versus Disk restore

3.2.4.3 Backup to Virtual Tape

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

Virtual Tape Library

- A virtual tape library (VTL) has the same components as that of a physical tape library except that the majority of the components are presented as virtual resources.
- For the backup software, there is no difference between a physical tape library and a virtual tape library.
- Fig 3.14 shows a virtual tape library that uses disks as backup media. Emulation software has a database with a list of virtual tapes, and each virtual tape is assigned a portion of a LUN on the disk. A virtual tape can span multiple LUNs if required.
- File system awareness is not required while backing up because virtual tape solutions use raw devices.
- Similar to a physical tape library, a robot mount is performed when a backup process starts in a virtual tape library. However, unlike a physical tape library, where this process involves some mechanical delays, in a virtual tape library it is almost instantaneous. Even the *load to ready* time is much less than in a physical tape library.
- After the virtual tape is mounted and the tape drive is positioned, the virtual tape is ready to be used, and backup data can be written to it. Unlike a physical tape library, the virtual tape library is not constrained by the shoe shining effect.
- When the operation is complete, the backup software issues a rewind command and then the tape can be unmounted. This rewind is also instantaneous.
- The virtual tape is then unmounted, and the virtual robotic arm is instructed to move it back to a virtual slot.
- The steps to restore data are similar to those in a physical tape library, but the restore operation is instantaneous. Even though virtual tapes are based on disks, which provide random access, they still emulate the tape behavior.
- Virtual tape library appliances offer a number of features that are not available with physical tape libraries.
- Some virtual tape libraries offer *multiple emulation engines* configured in an active cluster configuration. An engine is a dedicated server with a customized operating system that makes physical disks in the VTL appear as tapes to the backup application. With this feature, one engine can pick up the virtual resources from another engine in the event of any failure.
- Replication over IP is available with most of the virtual tape library appliances. This feature enables virtual tapes to be replicated over an inexpensive IP network to a remote

site.

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

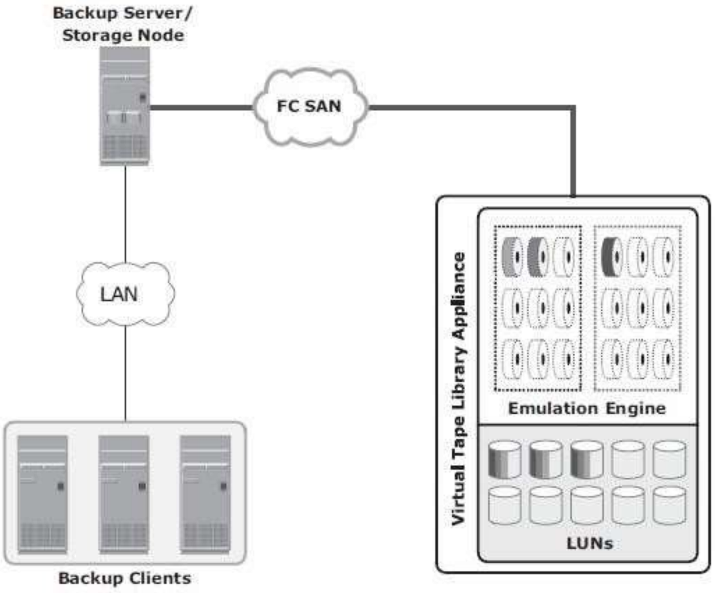
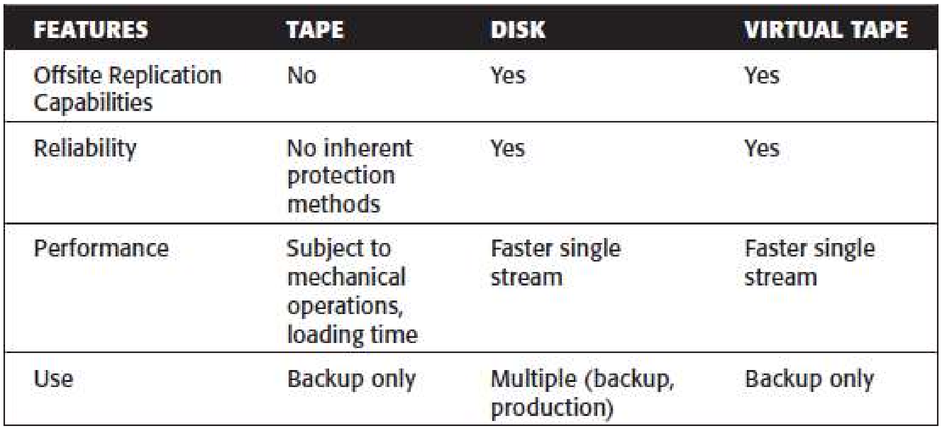


Fig 3.14: Virtual Tape Library

3.2.4.4 Backup Targets Comparison

Table 3.2 shows a comparison between various backup targets.



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

Table 3.2: Backup targets comparison

3.2.5 Data Deduplication for Backup

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

3.2.5.1 Data Deduplication Methods

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

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

3.2.5.2 Data Deduplication Implementation

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

Source-Based Data Deduplication

- *Source-based data deduplication* eliminates redundant data at the source before it transmits to the backup device.
- Source-based data deduplication can dramatically reduce the amount of backup data sent over the network during backup processes. It provides the benefits of a shorter backup window and requires less network bandwidth. There is also a substantial reduction in the capacity required to store the backup images.
- Fig 3.15 shows source-based data deduplication.
- Source-based deduplication increases the overhead on the backup client, which impacts the performance of the backup and application running on the client.
- Source-based deduplication might also require a change of backup software if it is not supported by backup software.

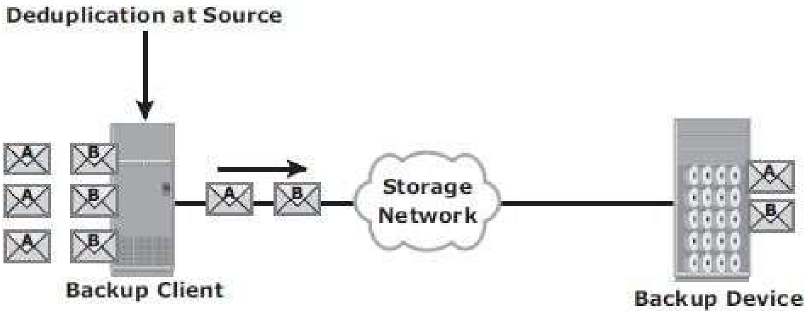


Fig 3.15: Source-based data deduplication

Target-Based Data Deduplication

- Target-based data deduplication is an alternative to source-based data deduplication.
- Target-based data deduplication occurs at the backup device, which offloads the backup client from the deduplication process.
- Fig 3.16 shows target-based data deduplication.
- In this case, the backup client sends the data to the backup device and the data is deduplicated at the backup device, either *immediately (inline)* or at a *scheduled time (post-process).*
- Because deduplication occurs at the target, all the backup data needs to be transferred over the network, which increases network bandwidth requirements. Target-based data deduplication does not require any changes in the existing backup software.
- *Inline deduplication* performs deduplication on the backup data before it is stored on the backup device. Hence, this method reduces the storage capacity needed for the backup.
- Inline deduplication introduces overhead in the form of the time required to identify and remove duplication in the data. So, this method is best suited for an environment with a large backup window.
- *Post-process deduplication* enables the backup data to be stored or written on the backup device first and then deduplicated later.
- This method is suitable for situations with tighter backup windows. However, post-process deduplication requires more storage capacity to store the backup images before they are deduplicated.

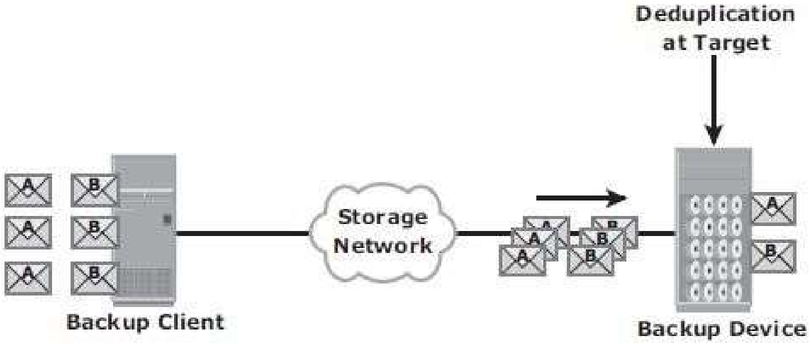


Fig 3.16: Target-based data deduplication

3.2.6 Backup in Virtualized Environments

- There are two approaches for performing a backup in a virtualized environment: the *traditional backup* approach and the *image-based* backup approach.
- In *the traditional backup* approach, a backup agent is installed either on the virtual machine (VM) or on the hypervisor.
- Fig 3.17 shows the traditional VM backup approach.
- If the backup agent is installed on a VM, the VM appears as a physical server to the agent. The backup agent installed on the VM backs up the VM data to the backup device. The agent does not capture VM files, such as the virtual BIOS file, VM swap file, logs, and configuration fi les. Therefore, for a VM restore, a user needs to manually re-create the VM and then restore data onto it.
- If the backup agent is installed on the hypervisor, the VMs appear as a set of files to the agent. So, VM files can be backed up by performing a file system backup from a hypervisor. This approach is relatively simple because it requires having the agent just on the hypervisor instead of all the VMs.
- The traditional backup method can cause high CPU utilization on the server being backed up.
- So the backup should be performed when the server resources are idle or during a low activity period on the network.
- And also allocate enough resources to manage the backup on each server when a large number of VMs are in the environment.

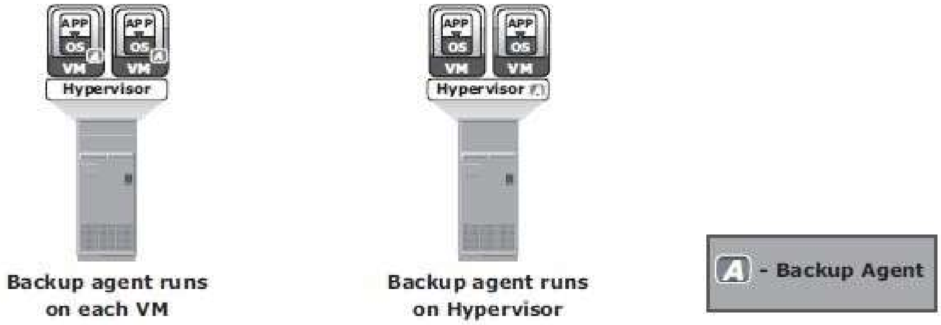


Fig 3.17: Traditional VM backup

- *Image-based backup* operates at the hypervisor level and essentially takes a snapshot of the VM.
- It creates a copy of the guest OS and all the data associated with it (snapshot of VM disk files), including the VM state and application configurations. The backup is saved as a single file called an *"image,"* and this image is mounted on the separate physical machine–proxy server, which acts as a backup client.
- The backup software then backs up these image files normally. (see Fig 3.18).
- This effectively offloads the backup processing from the hypervisor and transfers the load on the proxy server, thereby reducing the impact to VMs running on the hypervisor.
- Image-based backup enables quick restoration of a VM.

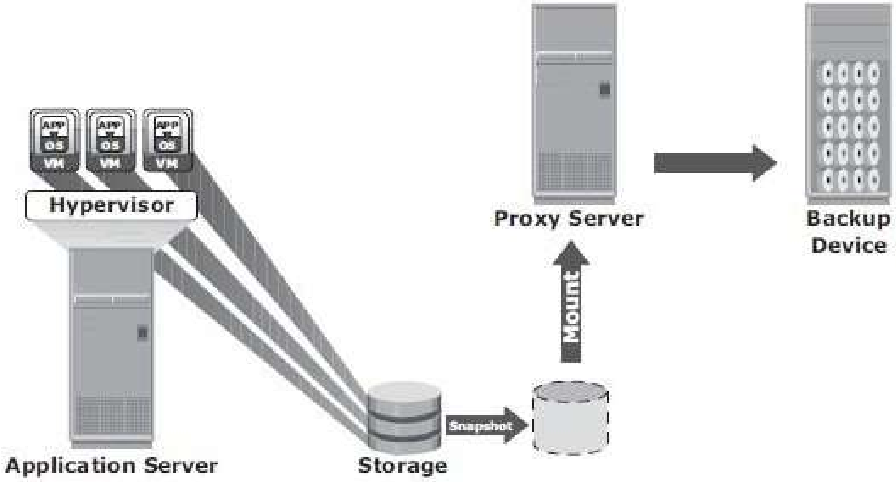


Fig 3.18: Image-based backup

3.3 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.

3.3.1 Source and Target

- A host accessing 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.
- A LUN on which the data is replicated is called the target LUN or simply the target or replica.
- Targets can also be accessed by hosts other than production hosts to perform operations such as backup or testing. Target data can be updated by the hosts accessing it without modifying the source.

3.3.2 Uses of Local Replicas

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.
- 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 the replica.

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 production device.

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 source volumes.

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 larger LUN.

3.3.3 Local Replication Technologies

- Local replication refers to replicating data within the same array or the same data center.
- Host-based and storage-based replications are the two major technologies adopted for local replication.

- File system replication and LVM-based replication are examples of host-based local replication technology.
- Storage array–based replication can be implemented with distinct solutions namely, full-volume mirroring, pointer based full-volume replication, and pointer-based virtual replication.

3.3.3.1 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
- 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 added or removed dynamically.

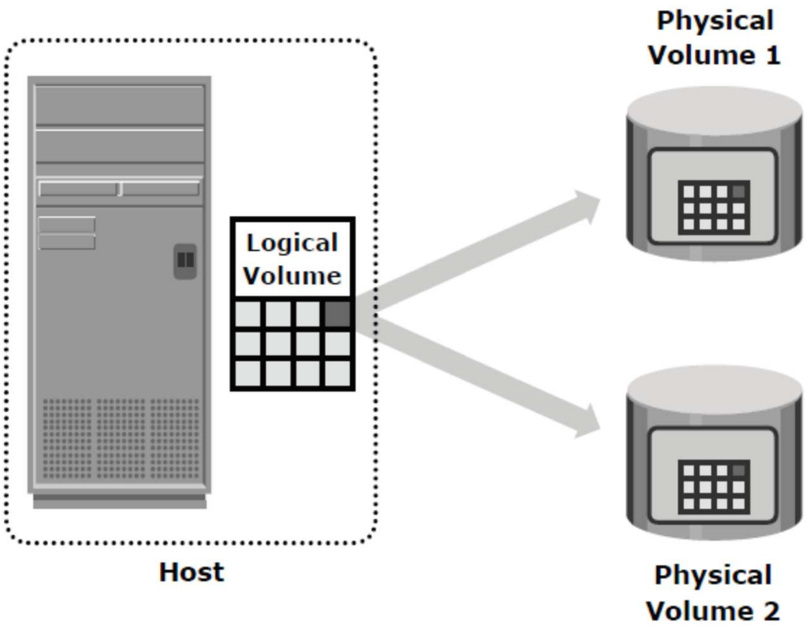


Fig 3.19: LVM-based mirroring

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.
- Tracking changes to the mirrors and performing incremental synchronization operations is also a challenge as all LVMs do not support incremental resynchronization.
- If the devices are already protected by some level of RAID on the array, then the additional protection provided by mirroring is unnecessary.
- This solution does not scale to provide replicas of federated databases and applications. Both the

replica and the source are stored within the same volume group. Therefore, the replica itself may become unavailable if there is an error in the volume group.

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

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.
- 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 data is read from that address.

3.3.3.2 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 3.20 shows storage array based local replication, where source and target are in the same array and accessed by different hosts.
- Storage array-based local replication can be further categorized as full-volume mirroring, pointer-based full-volume replication, and pointer-based virtual replication.
- Replica devices are also referred as target devices, accessible by business continuity host.

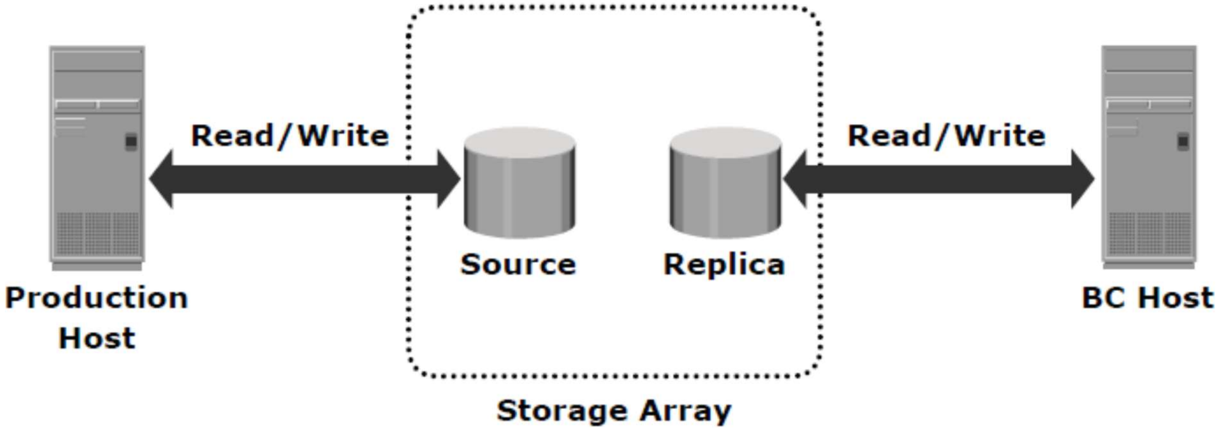


Fig 3.20: Storage array-based replication

Full-Volume Mirroring

- In full-volume mirroring, the target is attached to the source and established as a mirror of the source Figure 3.21(a).
- Existing data on the source is copied to the target. New updates to the source are also updated

on the target. After all the data is copied and both the source and the target contain identical data, the target can be considered 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 3.21(b) shows full-volume mirroring when the target is detached from the source.
- Both the source and the target can be accessed for read and write operations by the production hosts.

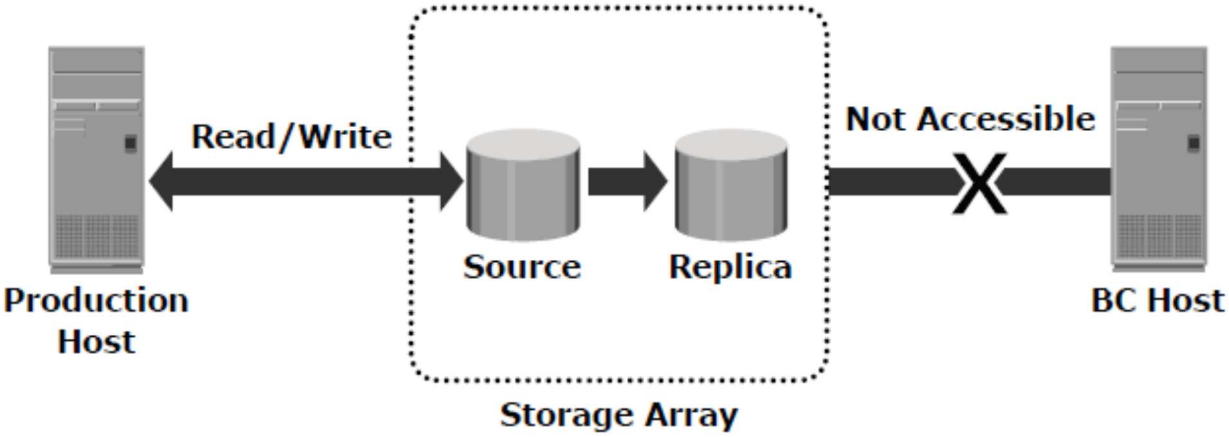


Fig 3.21(a): Full volume mirroring with source attached to replica

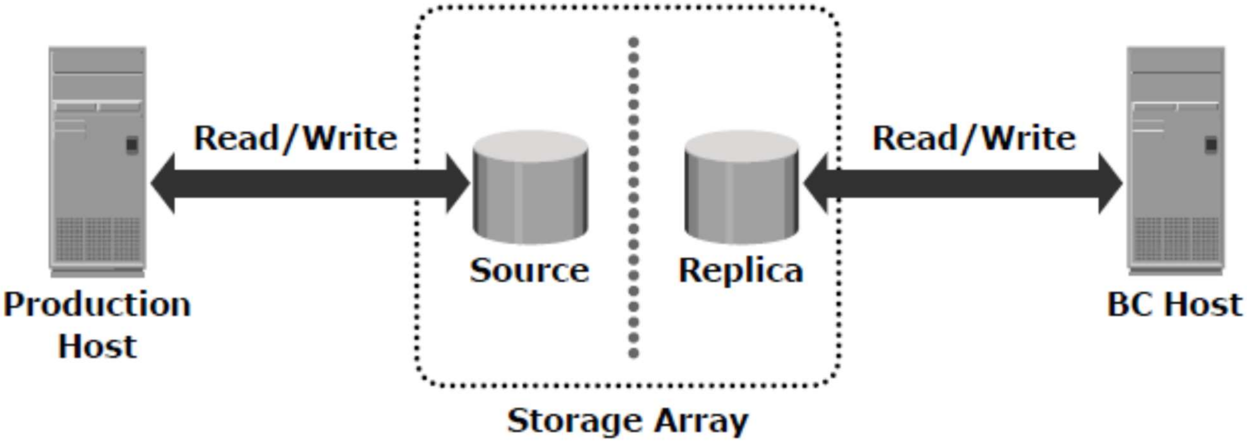


Fig 3.21(b): Full volume mirroring with source detached from replica

- 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 4:00 pm
- 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 these operations 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.

Pointer-Based, Full-Volume Replication

- Pointer-Based, Full-Volume Replication 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.
- 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:
- A write operation is issued to a specific address on the source for the first time (Figure 3.22).

- A read or write operation is issued to a specific address on the target for the first time(Figure 3.23 and Figure 3.24) .
- 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 original data at the point-in-time of activation is preserved on the target. This is illustrated in Figure 3.22.

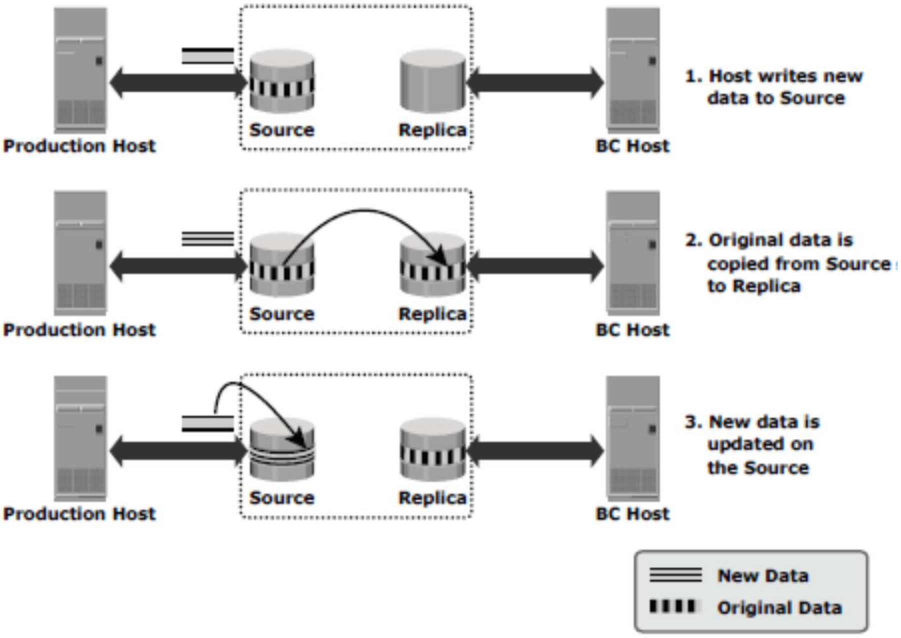

Production Host
Source
Replica
BC Host
1. Host writes new data to Source
2. Original data is copied from Source to Replica
3. New data is updated on the Source
New Data
Original Data

Figure 3.22: Copy on first access (CoFA) — write to source

- 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 availablet o the host. This is illustrated in Figure 3.23.

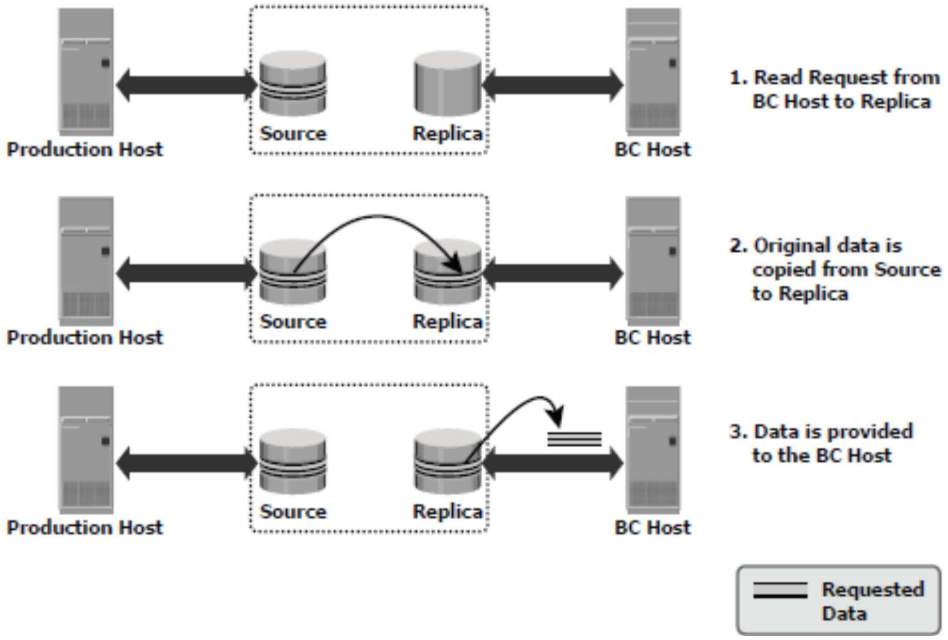

Production Host
Source
Replica
BC Host
1. Read Request from BC Host to Replica
2. Original data is copied from Source to Replica
3. Data is provided to the BC Host
Requested Data

Figure 3.23: 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 3.24.

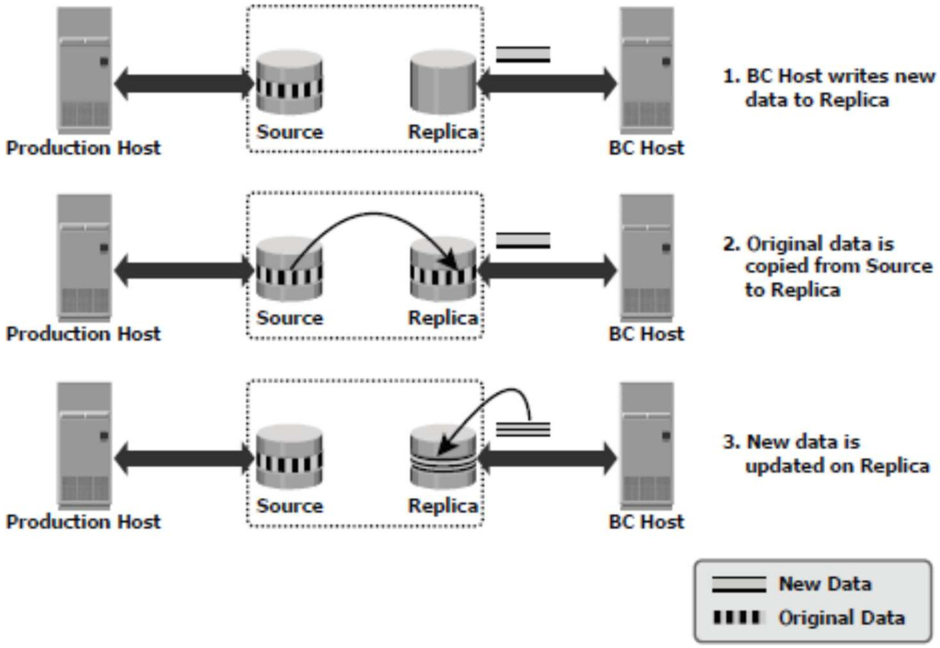

Production Host
Source
Replica
BC Host
1. BC Host writes new data to Replica
2. Original data is copied from Source to Replica
3. New data is updated on Replica
New Data
Original Data

Figure 3.24: Copy on first access (CoFA) — write to target

- 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 hence are termed Copy on First Access).
- If the replication session is terminated, then the target device only has the data that was accessed until the termination, not the entire contents of the source at the point-in-time. In this case, the data on the target cannot be used for a restore, as it is not a full replica of the source.
- In Full Copy mode, all data from the source is copied to the target in the background. Data is copied regardless of access. If access to a block that has not yet been copied is required, this block is preferentially copied to the target.
- In a complete cycle of the Full Copy mode, all data from the source is copied to the target. If the replication session is terminated now, the target will contain all the original data from the source at the point-in-time of activation. This makes the target a viable copy for recovery, restore, or other business continuity operations.
- The key difference between pointer-based, Full Copy mode and full-volume mirroring is that the target is immediately accessible on session activation in Full Copy mode. In contrast, one has to wait for synchronization and detachment to access the target in full-volume mirroring.
- Both the full-volume mirroring and pointer-based full-volume replication technologies require the target devices to be at least as large as the source devices.
- In addition, full-volume mirroring and pointer-based full-volume replication in Full Copy mode can provide incremental resynchronization or restore capability.

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 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 as shown in Figure 3.25.

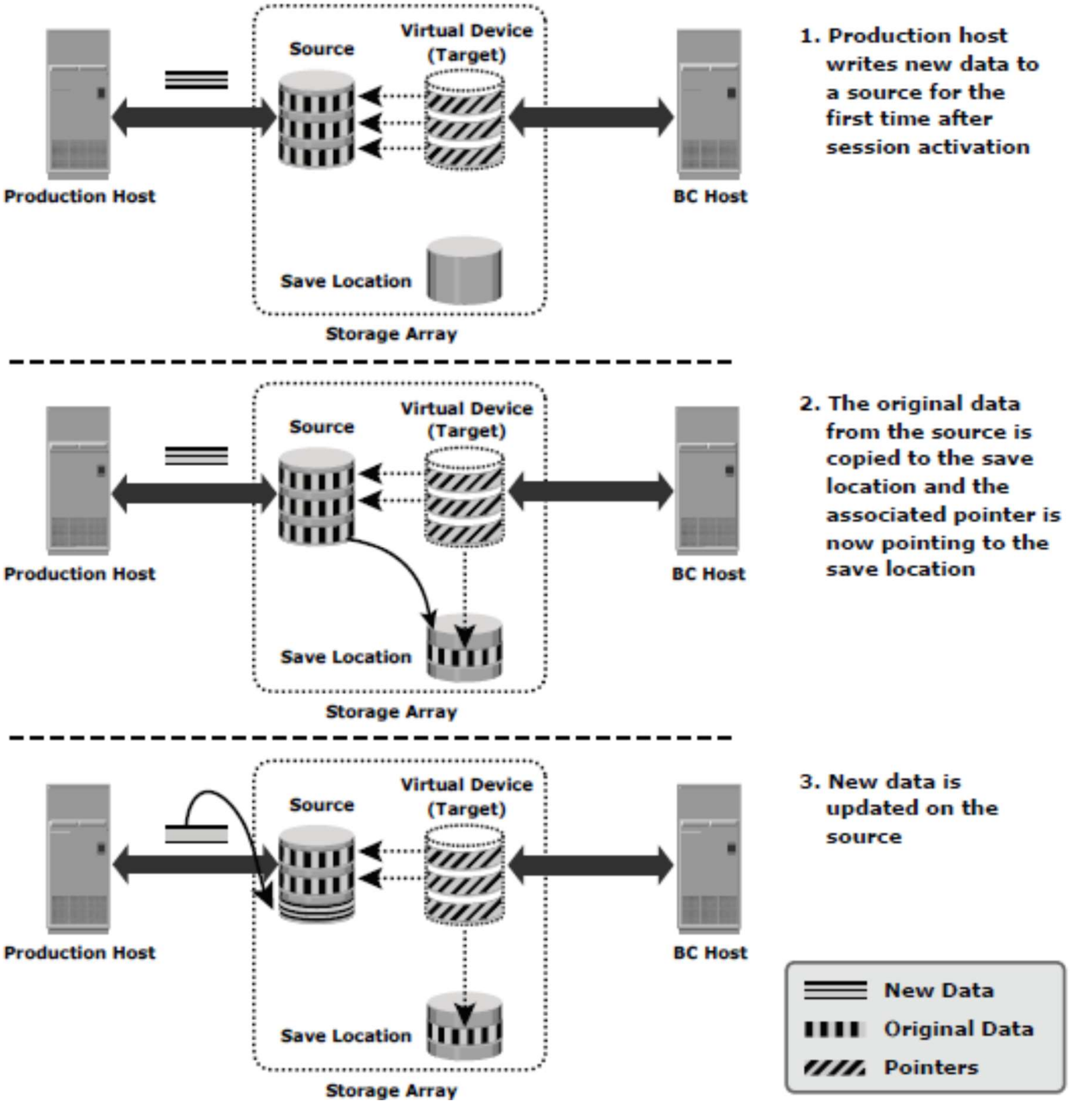

Source
Virtual Device (Target)
Production Host
BC Host
Save Location
Storage Array
1. Production host writes new data to a source for the first time after session activation
2. The original data from the source is copied to the save location and the associated pointer is now pointing to the save location
3. New data is updated on the source
New Data
Original Data
Pointers

Figure 3.25: Pointer-based virtual replication — write to source

- 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 as shown in Figure 3.26.

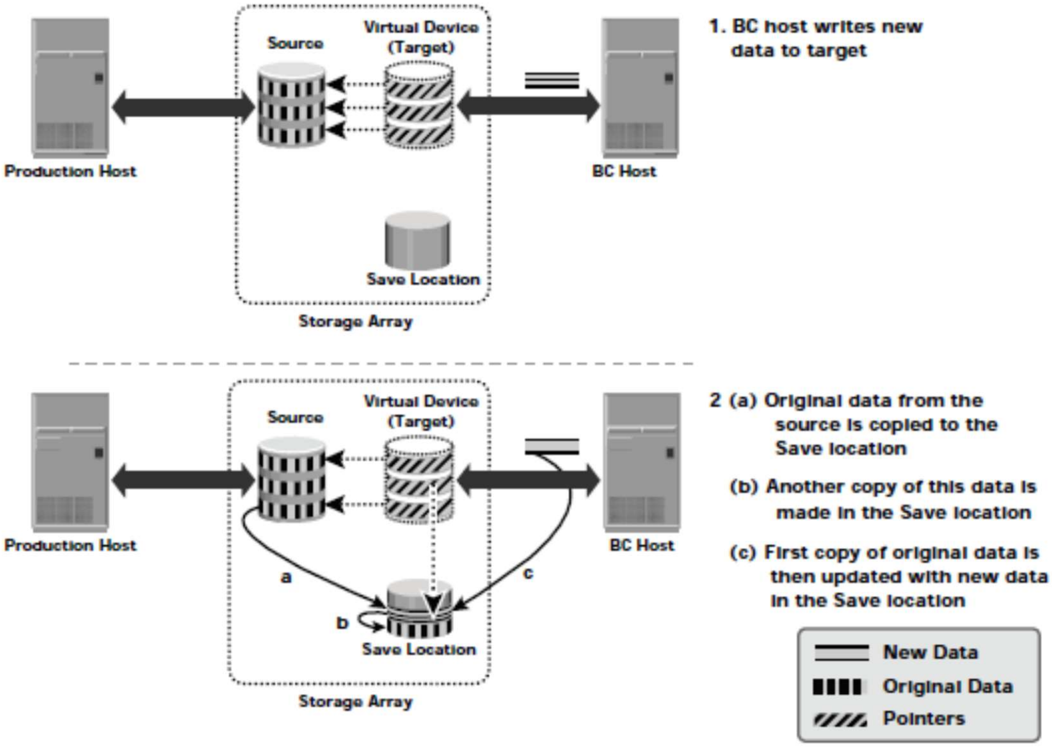

Source
Virtual Device (Target)
1. BC host writes new data to target
Production Host
BC Host
Save Location
Storage Array
Source
Virtual Device (Target)
2 (a) Original data from the source is copied to the Save location
(b) Another copy of this data is made in the Save location
(c) First copy of original data is then updated with new data in the Save location
Production Host
BC Host
a
b
c
Save Location
Storage Array
New Data
Original Data
Pointers

Figure 3.26: 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.

3.4 Remote Replication

- Remote replication is the process of creating replicas of information assets at remote sites (locations).
- Remote replicas help organizations mitigate the risks associated with regionally driven outages resulting from natural or human-made disasters.
- Similar to local replicas, they can also be used for other business operations.
- The infrastructure on which information assets are stored at the primary site is called the source. The infrastructure on which the replica is stored at the remote site is referred to as the target.
- Hosts that access the source or target are referred to as source hosts or target hosts, respectively.

3.4.1 Modes of Remote Replication

- The two basic modes of remote replication are synchronous and asynchronous.
- In synchronous remote replication, writes must be committed to the source and the target, prior to acknowledging "write complete" to the host (Figure 3.27).
- 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 the replica at all times.
- Further writes are transmitted to the remote site exactly in the order in which they are received at the source. Hence, write ordering is maintained.
- In the event of a failure of the source site, synchronous remote replication provides zero or near-zero RPO, as well as the lowest RTO.
- An application response time is increased with any synchronous remote replication. The degree of the impact on the response time depends on the distance between sites, available bandwidth, and the network connectivity infrastructure.
- The distances over which synchronous replication can be deployed depend on the application's ability to tolerate extension in response time. Typically, it is deployed for distances less than 200 KM (125miles) between the two sites.

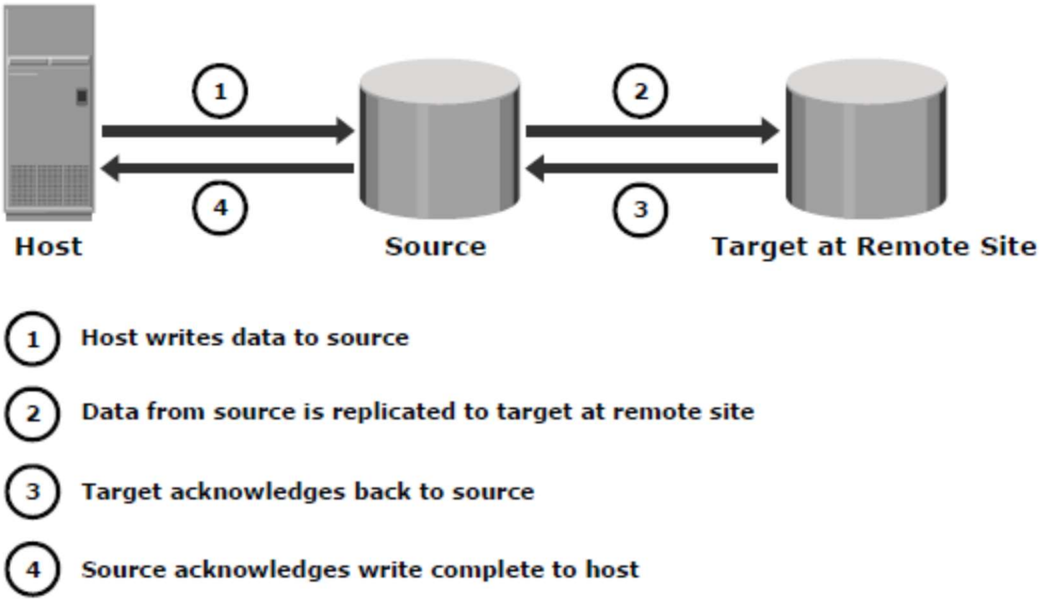

1
2
4
3
Host
Source
Target at Remote Site
1 Host writes data to source
2 Data from source is replicated to target at remote site
3 Target acknowledges back to source
4 Source acknowledges write complete to host

Figure 3.27: Synchronous replication

- In Asynchronous remote replication, 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 (Figure 3.28). This eliminates the impact to the application's response time.
- Data at the remote site will be behind the source by at least the size of the buffer. Hence, asynchronous remote replication provides a finite (nonzero) RPO disaster recovery solution.
- RPO depends on the size of the buffer, available network bandwidth, and the write workload to the source.
- There is no impact on application response time, as the writes are acknowledged immediately to the source host. This enables deployment of asynchronous replication over extended distances.
- Asynchronous remote replication can be deployed over distances ranging from several hundred to several thousand kilometers between two sites.

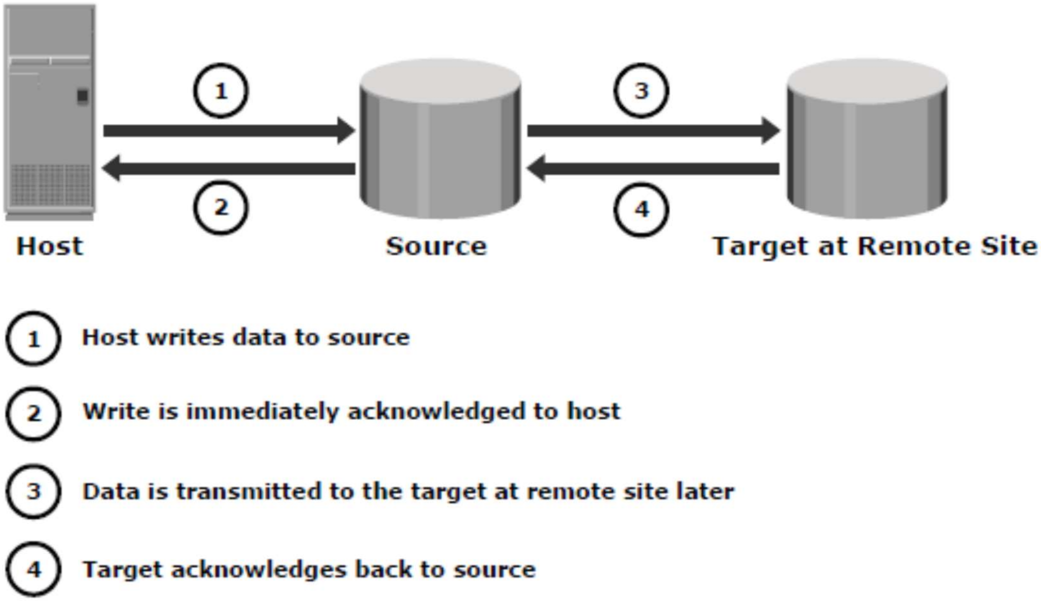

1
2
3
4
Host
Source
Target at Remote Site
1 Host writes data to source
2 Write is immediately acknowledged to host
3 Data is transmitted to the target at remote site later
4 Target acknowledges back to source

Figure 3.28: Asynchronous replication

3.4.2 Remote Replication Technologies

- Remote replication of data can be handled by the hosts or by the storage arrays.
- Other options include specialized appliances to replicate data over the LAN or the SAN, as well as replication between storage arrays over the SAN.

3.4.2.1 Host-Based Remote Replication

- Host-based remote replication uses one or more components of the host to perform and manage the replication operation.
- There are two basic approaches to host-based remote replication: LVM-based replication and database replication via log shipping.

LVM-Based Remote Replication

- LVM-based replication is performed and managed at the volume group level.
- Writes to the source volumes are transmitted to the remote host by the LVM.
- The LVM on the remote host receives the writes and commits them to the remote volume group.
- Prior to the start of replication, identical volume groups, logical volumes, and file systems are

created at the source and target sites.

- Initial synchronization of data between the source and the replica can be performed in a number of ways.
- One method is to backup the source data to tape and restore the data to the remote replica. Alternatively, it can be performed by replicating over the IP network.
- Until completion of initial synchronization, production work on the source volumes is typically halted. After initial synchronization, production work can be started on the source volumes and replication of data can be performed over an existing standard IP network (Figure 3.29).

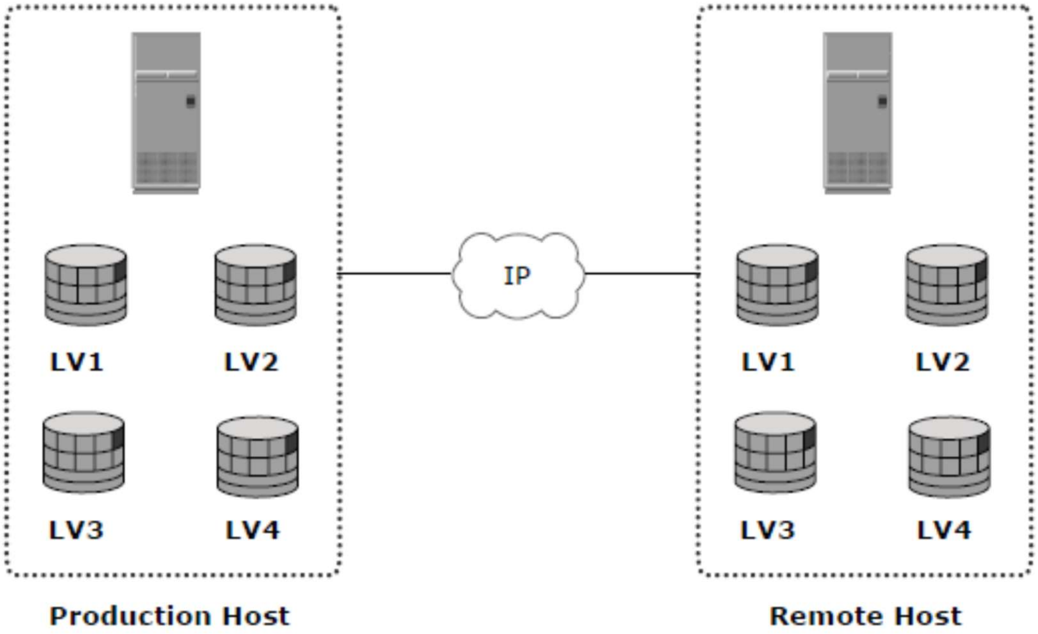


Figure 3.29: LVM-based remote replication

- LVM-based remote replication supports both synchronous and asynchronous modes of data transfer.
- In asynchronous mode, writes are queued in a log file at the source and sent to the remote host in the order in which they were received.
- The size of the log file determines the RPO at the remote site.
- In the event of a network failure, writes continue to accumulate in the log file. If the log file fills up before the failure is resolved, then a full resynchronization is required upon network availability.

- In the event of a failure at the source site, applications can be restarted on the remote host, using the data on the remote replicas.
- LVM-based remote replication eliminates the need for a dedicated SAN infrastructure.
- LVM-based remote replication is independent of the storage arrays and types of disks at the source and remote sites.
- Most operating systems are shipped with LVMs, so additional licenses and specialized hardware are not typically required.
- The replication process adds overhead on the host CPUs. CPU resources on the source host are shared between replication tasks and applications, which may cause performance degradation of the application.
- As the remote host is also involved in the replication process, it has to be continuously up and available.
- LVM-based remote replication does not scale well, particularly in the case of applications using federated databases

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 (Figure 3.30).
- The remote host receives the logs and applies them to the remote database.
- 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. Some implementations allow reads and writes from the standby database.
- 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 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, and rate of updates to the source database, as well as the frequency of log switching, should be considered when determining the optimal size of the log file.
- Because the source host does not transmit every update and buffer them, this alleviates the burden on the source host CPU.
- Similar to LVM-based remote replication, the existing standard IP network can be used for replicating log files. Host-based log shipping does not scale well, particularly in the case of applications using federated databases.

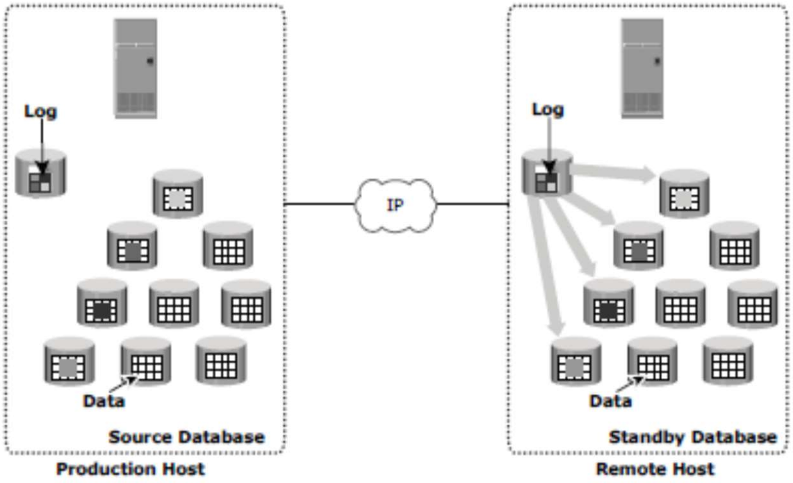


Figure 3.30: Host-based log shipping

3.4.2.1 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 utilized for running an application.
- 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.
- Three-site remote replication can be implemented using a combination of synchronous mode and asynchronous mode, as well as a combination of synchronous mode and disk-buffered mode.

Array based 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 host.
- Additional writes on that source cannot occur until each preceding write has been completed and acknowledged (Figure 3.31).
- In the case of synchronous 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 can de-stage 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 can keep track of the writes that are not transmitted to the remote storage array. When the network links are restored, the accumulated data can be transmitted to the remote storage array.
- During the time of network link outage, if there is a failure at the source site, some data will be lost and the RPO at the target will not be zero.

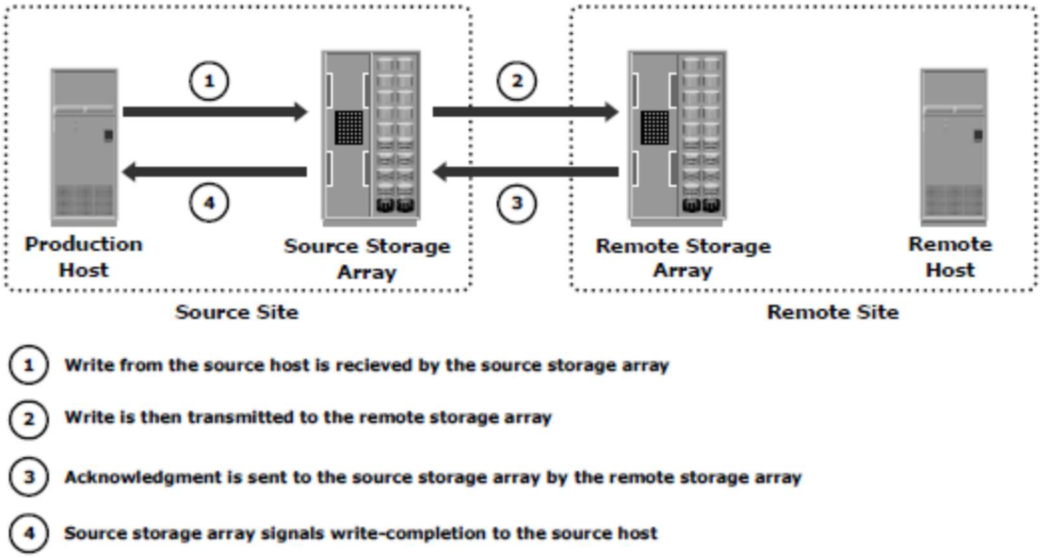

1
2
3
4
Production Host
Source Storage Array
Remote Storage Array
Remote Host
Source Site
Remote Site
1 Write from the source host is recieved by the source storage array
2 Write is then transmitted to the remote storage array
3 Acknowledgment is sent to the source storage array by the remote storage array
4 Source storage array signals write-completion to the source host

Figure 3.31: Array-based synchronous remote replication

- For synchronous remote replication, network bandwidth equal to or greater than the maximum write workload between the two sites should be provided at all times.
- Figure 3.32 illustrates the write workload (expressed in MB/s) over time. The "Max" line indicated in Figure 3.32 represents the required bandwidth that must be provisioned for synchronous replication. Bandwidths lower than the maximum write workload results in an unacceptable increase in application response time.

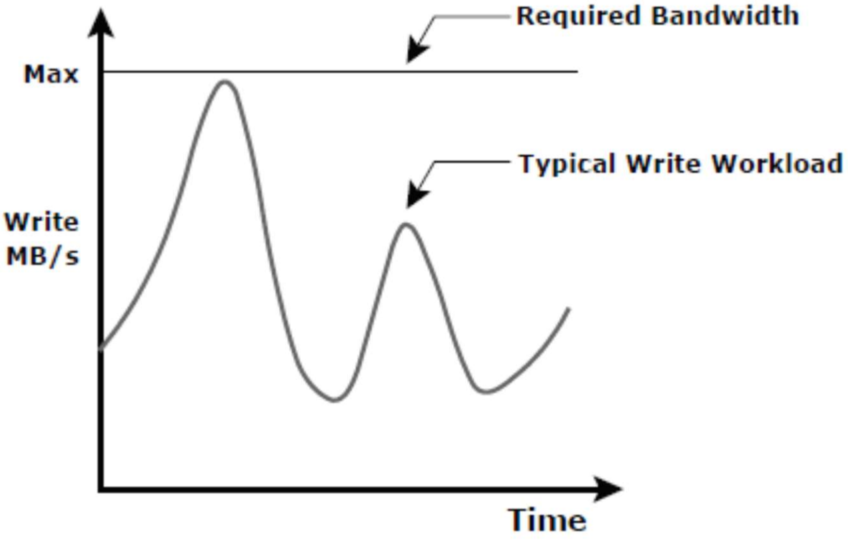
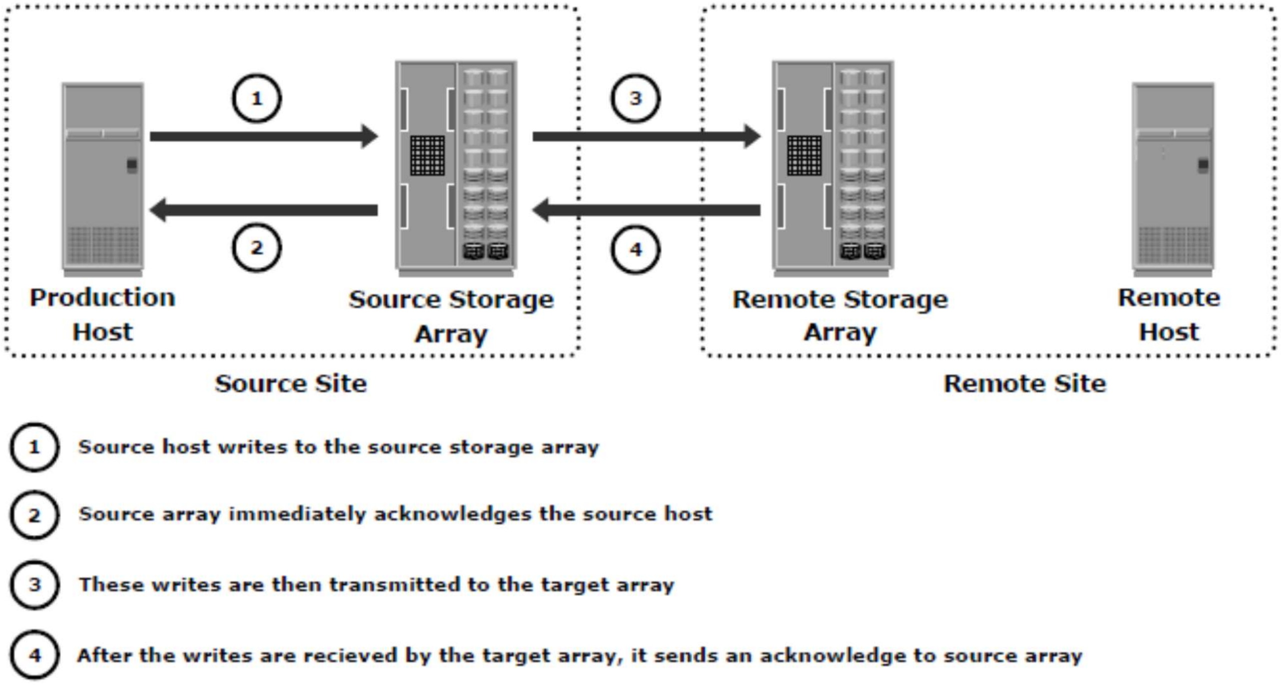


Figure 3.32: Network bandwidth requirement for synchronous replication

Array based Asynchronous Replication Mode

- ➢ In array-based asynchronous remote replication mode, 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 Source host writes to the source storage array

2 Source array immediately acknowledges the source host

3 These writes are then transmitted to the target array

4 After the writes are recieved by the target array, it sends an acknowledge to source array

Figure 3.33: Array-based asynchronous remote replication

- ➢ Similar to synchronous replication, asynchronous replication writes are placed in cache on the two arrays and are later de-staged to the appropriate disks.
- ➢ Some implementations of asynchronous remote replication maintain write ordering. A time stamp 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 to most DBMSs.

- 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, as only bandwidth equal to or greater than the average write workload is needed, as represented by the "Average" line in Figure 3.34.
- During times when the write workload exceeds the average bandwidth, sufficient buffer space has to be configured on the source storage array to hold these writes.

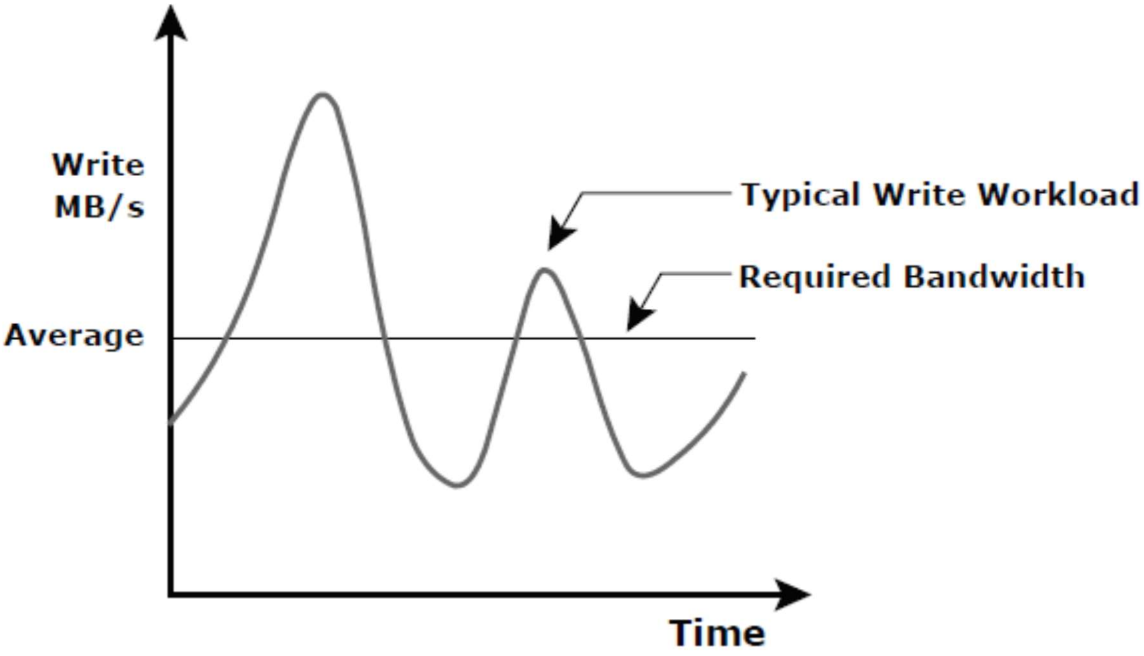


Figure 3.34: Network bandwidth requirement for asynchronous replication

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.
- The sequence of operations in a disk-buffered remote replication is shown in Figure 3.35.
- 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 is running 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 is transmitted 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 available link bandwidth and the data change rate on the source device.

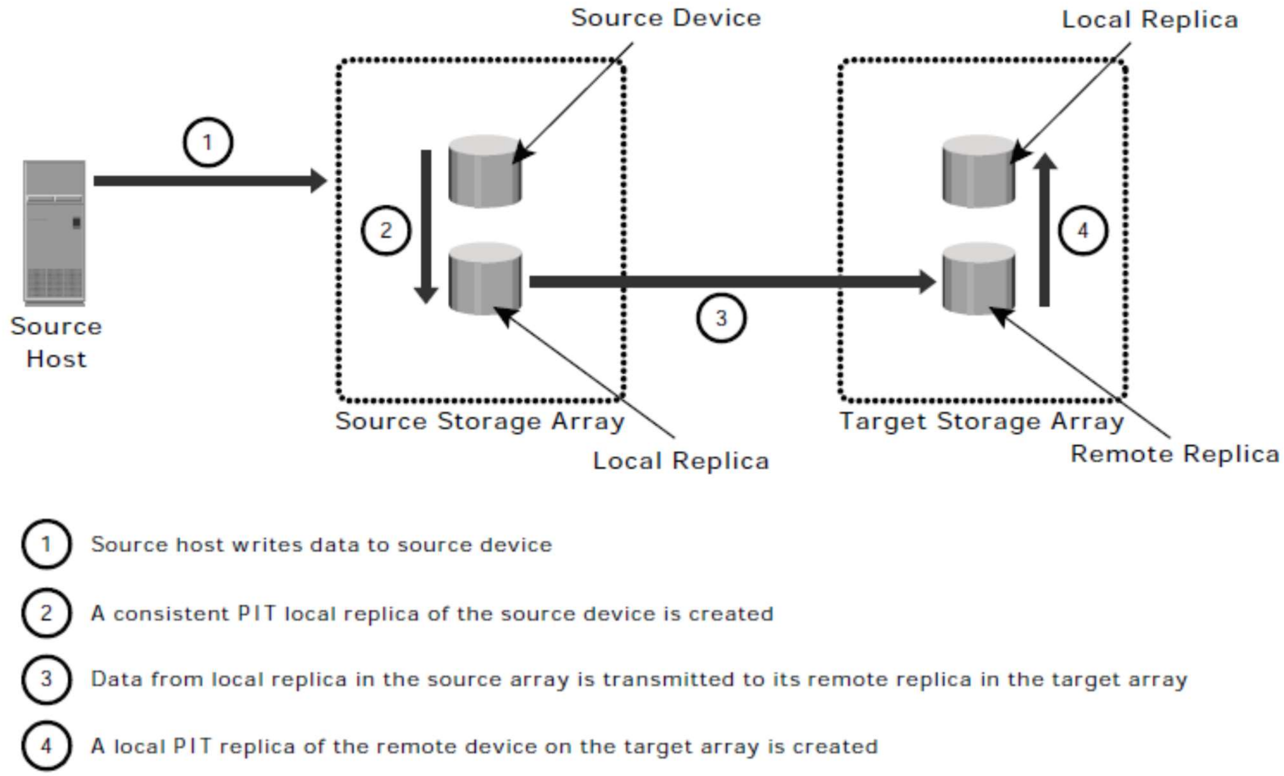

Source Device
Local Replica
1
2
3
4
Source Host
Source Storage Array
Local Replica
Target Storage Array
Remote Replica
1 Source host writes data to source device
2 A consistent PIT local replica of the source device is created
3 Data from local replica in the source array is transmitted to its remote replica in the target array
4 A local PIT replica of the remote device on the target array is created

Figure 3.35: Disk-buffered remote replication

- Array-based replication technologies can track changes made to the source and target devices. Hence, all resynchronization operations can be done incrementally.
- For example, a local replica of the source device is created at 10:00 am and this data is transmitted to the remote replica, which takes one hour to complete. Changes made to the source device after 10:00 am are tracked.
- Another replica of the source device is created at 11:00 am by applying track changes between

the source and local replica (10:00 am copy).

- During the next cycle of transmission (11:00 am data), the source data has moved to 12:00 pm The local replica in the remote array has the 10:00 am data until the 11:00 am 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 two hours (as the remote site has 10:00 am data).

Three-Site Replication

- In synchronous and asynchronous replication, under normal conditions the workload is running at the source site. Operations at the source site will not be disrupted by any failure to the target site or to the network used for replication. The replication process resumes as soon as the link or target site issues are resolved.
- The source site continues to operate without any remote protection. If failure occurs at the source site during this time, RPO will be extended.
- In synchronous replication, source and target sites are usually within 200 KM of each other. Hence, in the event of a regional disaster, both the source and the target sites could become unavailable. This will lead to extended RPO and RTO because the last known good copy of data would have to come from another source, such as offsite tape library.
- A regional disaster will not affect the target site in asynchronous replication, as 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 will be no remote protection until the failure is resolved.
- Three-site replication is used to mitigate the risks identified in two-site replication.
- In a three-site replication, data from the source site is replicated to two remote data centers.
- Replication can be synchronous to one of the two data centers, providing a zero-RPO solution. It can be asynchronous or disk buffered to the other remote data center, providing a finite RPO.
- Three-site remote replication can be implemented as a cascade/multi-hop or a triangle/multi-target solution.

Three-Site Replication—Cascade/Multi-hop

- In the cascade/multi-hop form of 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 bunker occurs synchronously, but replication between the bunker and the remote site can be achieved in two ways: 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 the asynchronous replication to create a remote replica at the remote site(Figure 3.36)

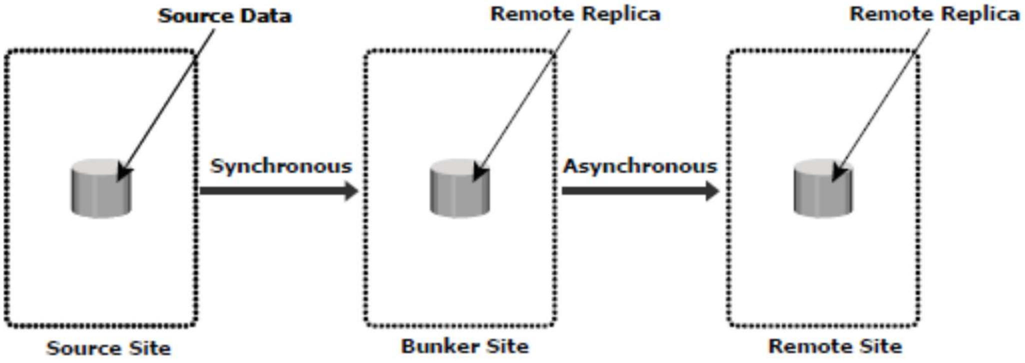


Figure 3.36: Synchronous + asynchronous three site replication

- RPO at the remote site is usually on the order of minutes in this implementation. In this method, a minimum of three storage devices are required (including the source) to replicate one storage device.
- The devices containing a synchronous remote replica at the bunker and the asynchronous replica at the remote are the other two devices.
- If there is a disaster at the source, 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 on the order of minutes.
- 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 will continue to operate as normal but without any remote replication.
- This situation is very similar to two-site replication when a failure/disaster occurs at the target site. The updates to the remote site cannot occur due to the failure in the bunker site. Hence, the data at the remote site keeps falling behind, but the advantage here is that if the source fails during this time, operations can be resumed at the remote site.
- RPO at the remote site depends on the time difference between the bunker site failure and source site failure.
- A regional disaster in three-site cascade/multihop replication is very similar to a source site failure in two-site asynchronous replication. Operations will failover to the remote site with an RPO on 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 3.37 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 in this implementation.
- For example, if a local replica is created at 10:00 am at the bunker and it takes an hour to transmit this data to the remote site, changes made to the remote replica at the bunker since 10:00 am are tracked. Hence only one hour's worth of data has to be resynchronized between the bunker and the remote site during the next cycle. RPO in this case will also be two hours, similar to disk-buffered replication.
- The process of creating the consistent PIT copy at the bunker and incrementally updating the remote replica and the local replica at the remote site occurs continuously in a cycle. This process can be automated and controlled from the source.

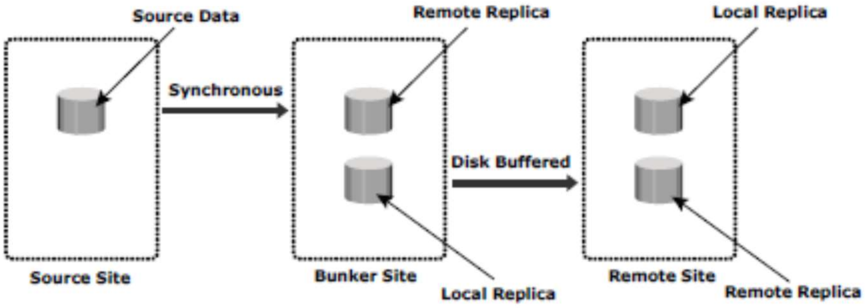


Figure 3.37: Synchronous + Disk buffered three site replication

Three-Site Replication—Triangle/Multi-target

- In the three-site triangle/multi-target replication, data at the source storage array is concurrently replicated to two different arrays.
- The source-to-bunker site (target 1) replication is synchronous, with a near-zero RPO. The source-to remote site (target 2) replication is asynchronous, with an RPO of minutes.
- The distance between the source and the remote site could be thousands of miles. This configuration 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 key benefit of three-site triangle/multi-target 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 them. Resynchronization between the two surviving target sites is incremental. Disaster recovery protection is always available in the event of any one site failure.

- During normal operations all three sites are available and the 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 in the event of a source site disaster, operations can be resumed at the bunker or the remote sites with incremental resynchronization between the sites.

SAN-Based Remote Replication

- SAN-based remote replication enables the replication of data between heterogeneous storage arrays.
- Data is moved from one array to the other over the SAN/ WAN.
- This technology is application and server operating system independent, because the replication operations are performed by one of the storage arrays (the control array). There is no impact on production servers (because replication is done by the array) or the LAN (because data is moved over the SAN).
- SAN-based remote replication is a point-in-time replication technology.
- Uses of SAN-based remote replication include data mobility, remote vaulting, and data migration.
- Data mobility enables incrementally copying multiple volumes over extended distances, as well as implementing a tiered storage strategy.
- Data vaulting is the practice of storing a set of point-in-time copies on heterogeneous remote arrays to guard against a failure of the source site.
- Data migration refers to moving data to new storage arrays and consolidating data from multiple heterogeneous storage arrays onto a single storage array.

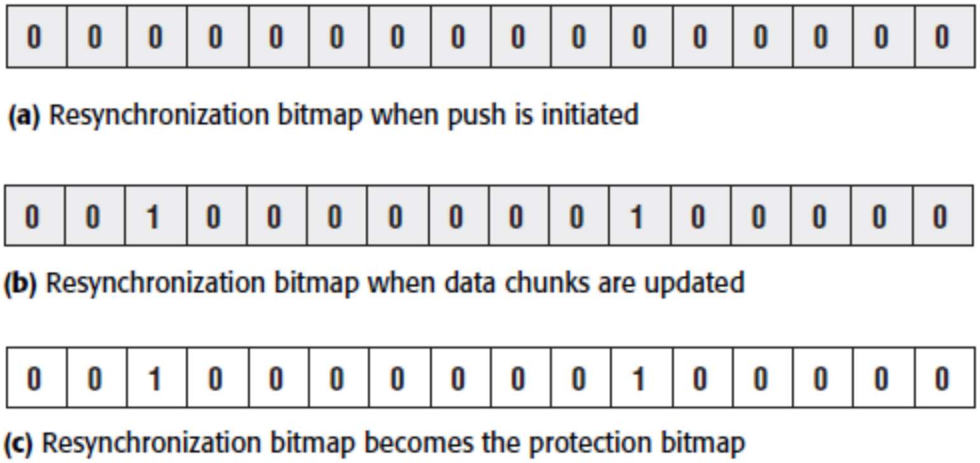
- The array performing the replication operations is called the control array. Data can be moved to/from devices in the control array to/from a remote array.
- The devices in the control array that are part of the replication session are called control devices. For every control device there is a counterpart, a remote device, on the remote array.
- The terms “control” or “remote” do not indicate the direction of data flow, they only indicate the array that is performing the replication operation. Data movement could be from the control array to the remote array or vice versa. The direction of data movement is determined by the replication operation.
- The front-end ports of the control array must be zoned to the front-end ports of the remote array. LUN masking should be performed on the remote array to allow access to the remote devices to the front-end port of the control array.
- In effect, the front-end ports of the control array act as an HBA, initiating data transfer to/from the remote array.
- SAN-based replication uses two types of operations: push and pull.
- In the push operation, data is transmitted from the control storage array to the remote storage array. The control device, therefore, acts like the source, while the remote device is the target. The data that needs to be replicated would be on devices in the control array
- In the pull operation, data is transmitted from the remote storage array to the control storage array. The remote device is the source and the control device is the target. The data that needs to be replicated would be on devices in the remote array.
- When a push or pull operation is initiated, the control array creates a protection bitmap to track the replication process. Each bit in the protection bitmap represents a data chunk on the control device. Chunk size may vary with technology implementations.
- When the replication operation is initiated, all the bits are set to one, indicating that all the contents of the source device need to be copied to the target device. As the replication process copies data, the bits are changed to zero, indicating that a particular chunk has been copied. At the end of the replication process, all the bits become zero.
- During the push and pull operations, host access to the remote device is not allowed because the control storage array has no control over the remote storage array and cannot track any change on the remote device. Data integrity cannot be guaranteed if changes are made to the remote

device during the push and pull operations.

- Therefore, for all SAN-based remote replications, the remote devices should not be in use during the replication process in order to ensure data integrity and consistency.
- The push/pull operations can be either hot or cold.
- In a cold operation, the control device is inaccessible to the host during replication. Cold operations guarantee data consistency because both the control and the remote devices are offline to every host operation.
- In a hot operation, the control device is online for host operations. With hot operations, changes can be made to the control device during push/pull because the control array can keep track of all changes, and thus ensures data integrity.
- When the hot push operation is initiated, applications can be up and running on the control devices.
- I/O to the control devices is held while the protection bitmap is created. This ensures a consistent PIT image of the data. The protection bitmap is referred prior to any write to the control devices. If the bit is zero, the write is allowed. If the bit is one, the replication process holds the write, copies the required chunk to the remote device, and then allows the write to complete
- In the hot pull operation, the hosts can access control devices after starting the pull operation. The protection bitmap is referenced for every read or write operation. If the bit is zero, a read or write occurs. If the bit is one, the read or write is held, and the replication process copies the required chunk from the remote device. When the chunk is copied, the read or write is completed.
- The control devices can be used after the pull operation is initiated and as soon as the protection bitmap is created.
- In SAN-based replication, the control array can keep track of changes made to the control devices after the replication session is activated. This is allowed in the incremental push operation only. A second bitmap, called a resynchronization bitmap, is created. All the bits in the resynchronization bitmap are set to zero when a push is initiated, as shown in Figure 3.38 (a).
- As changes are made to the control device, the bits are flipped from zero to one, indicating that

changes have occurred, as shown in Figure 3.38 (b).

- When resynchronization is required, the push is reinitiated and the resynchronization bitmap becomes the new protection bitmap, as shown in Figure 3.38(c), and only the modified chunks are transmitted to the remote devices.
- If changes are made to the remote device, the SAN-based replication operation is unaware of these changes, therefore, data integrity cannot be ensured if an incremental push is performed.



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a) Resynchronization bitmap when push is initiated

0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

(b) Resynchronization bitmap when data chunks are updated

0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

(c) Resynchronization bitmap becomes the protection bitmap

Figure 3.38: Bitmap status in SAN-based replication