Introduction to Business Continuity

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.

Information Availability:

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

IA can be defined in terms of accessibility, reliability, and timeliness of information.

- **Accessibility:** Information should be accessible at the right place, to the right user.
- **Reliability:** Information should be reliable and correct in all aspects. It is "the same" as what was stored, and there is no alteration or corruption to the information.
- **Timeliness:** Defines the exact moment or the time window (a particular time of the day, week, month, and year as specified) during which information must be accessible.

For example, if online access to an application is required between 8:00 a.m. and 10:00 p.m. each day, any disruptions to data availability outside of this time slot are not considered to affect timeliness.

Causes of Information Unavailability:

Various planned and unplanned incidents result in information unavailability. Planned outages include installation/integration/maintenance of new hardware, software upgrades or patches, taking backups, application and data restores, facility operations (renovation and construction), and refresh/migration of the testing to the production environment.

Unplanned outages include failure caused by human errors, database corruption, and failure of physical and virtual components.

Another type of incident that may cause data unavailability is natural or manmade disasters, such as flood, fire, earthquake, and contamination.

As illustrated in Figure 9-1, the majority of outages are planned.

Planned outages are expected and scheduled but still cause data to be unavailable.

Statistically, the cause of information unavailability due to unforeseen disasters is less than 1 percent.

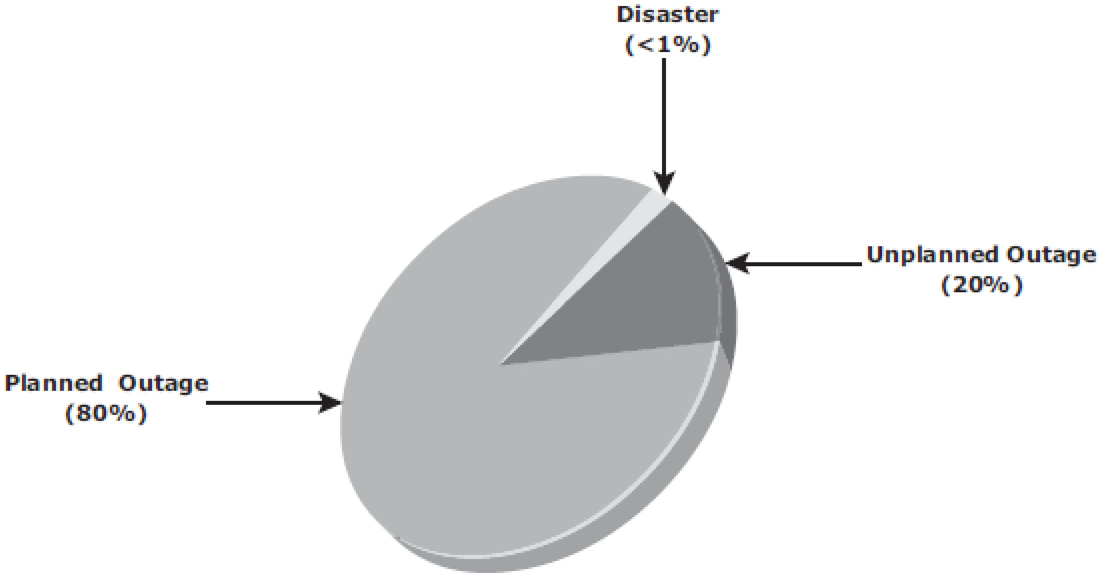


Figure 9-1: Disruptors of information availability

Consequences of Downtime:

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

An important metric, average cost of downtime per hour, provides a key estimate in determining the appropriate BC solutions. It is calculated as follows:

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

Where:

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

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

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

Measuring Information Availability:

IA relies on the availability of both physical and virtual components of a data center. Failure of these components might disrupt IA.

A failure is the termination of a component’s capability to perform a required function. The component’s capability can be restored by performing an external corrective action, such as a manual reboot, repair, or replacement of the failed component(s).

Repair involves restoring a component to a condition that enables it to perform a required function.

Proactive risk analysis, performed as part of the BC planning process, considers the component failure rate and average repair time, which are measured by mean time between failure (MTBF) and mean time to repair (MTTR):

Mean Time Between Failure (MTBF): It is the average time available for a system or component to perform its normal operations between failures. It is the measure of system or component reliability and is usually expressed in hours.

Mean Time To Repair (MTTR): It is the average time required to repair a failed component. While calculating MTTR, it is assumed that the fault responsible for the failure is correctly identified and the required spares and personnel are available. A fault is a physical defect at the component level, which may result in information unavailability.

MTTR includes the total time required to do the following activities: Detect the fault, mobilize the maintenance team, diagnose the fault, obtain the spare parts, repair, test, and restore the data.

Figure 9-2 illustrates the various information availability metrics that represent system uptime and downtime.

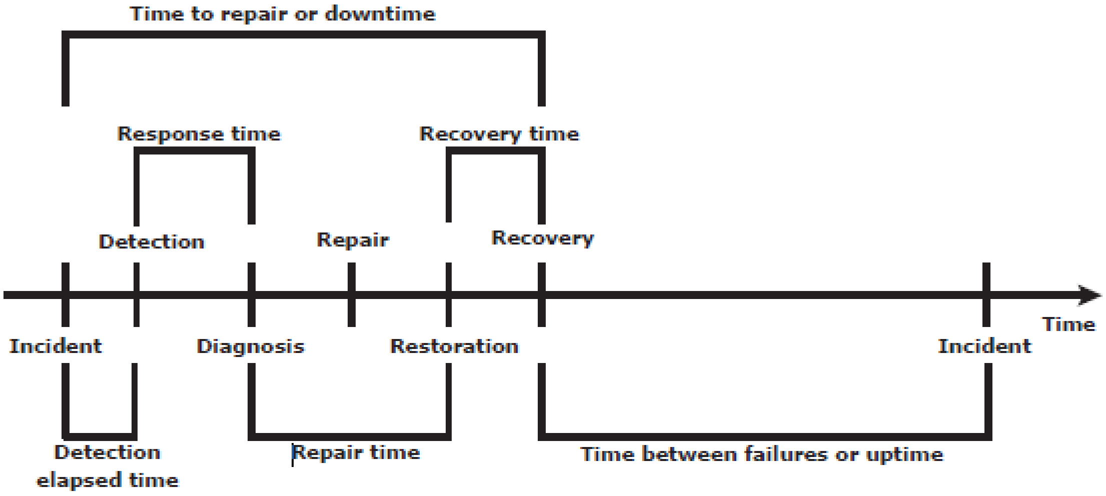


Figure 9-2: Information availability metrics

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

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

Where system uptime is the period of time during which the system is in an accessible state; when it is not accessible, it is termed as system downtime.

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

IA = MTBF/(MTBF + MTTR)

Uptime per year is based on the exact timeliness requirements of the service.

This calculation leads to the number of "9s" representation for availability metrics. Table 9-1 lists the approximate amount of downtime allowed for a service to achieve certain levels of 9s availability.

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

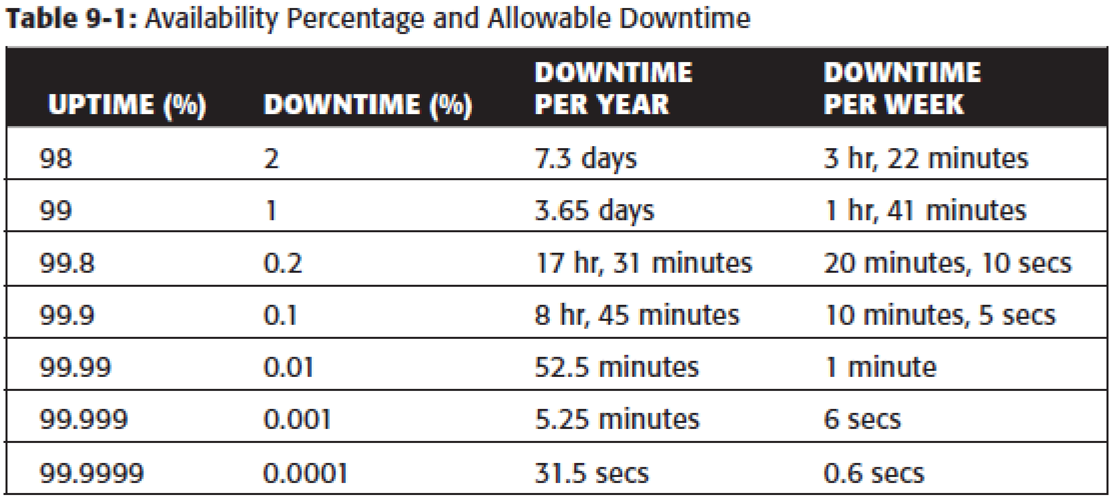


Table 9-1: Availability Percentage and Allowable Downtime

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

BC Terminology:

- **Disaster recovery:** This is the coordinated process of restoring systems, data, and the infrastructure required to support ongoing business operations after a disaster occurs. It is the process of restoring a previous copy of the data and applying logs or other necessary processes to that copy to bring it to a known point of consistency. After all recovery efforts are completed, the data is validated to ensure that it is correct.
- **Disaster restart:** This is the process of restarting business operations with mirrored consistent copies of data and applications.
- **Recovery-Point Objective (RPO):** This is the point in time to which systems and data must be recovered after an outage. It defines the amount of data loss that a business can endure. A large RPO signifies high tolerance to information loss in a business. Based on the RPO, organizations plan for the frequency with which a backup or replica must be made.

  For example, if the RPO is 6 hours, backups or replicas must be made at least once in 6 hours. Figure 9-3 (a) shows various RPOs and their corresponding ideal recovery strategies.

  An organization can plan for an appropriate BC technology solution on the basis of the RPO it sets.

  For example:

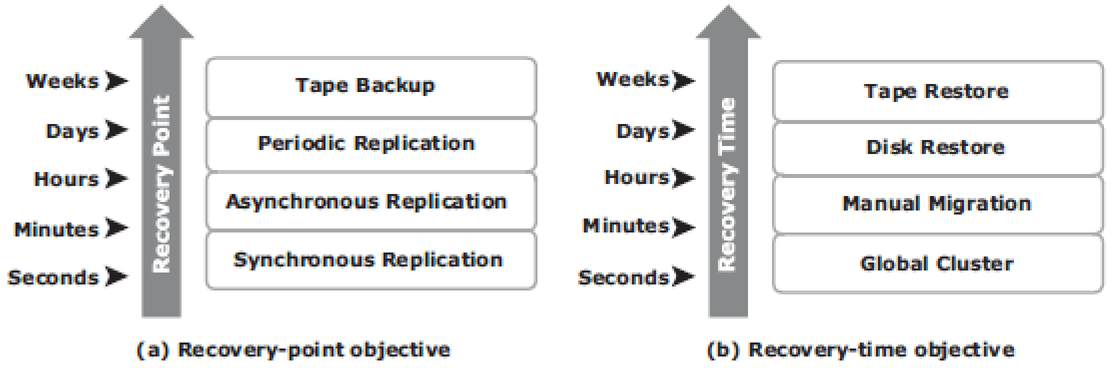
  - **RPO of 24 hours:** Backups are created at an offsite tape library every midnight. The corresponding recovery strategy is to restore data from the set of last backup tapes.

- **RPO of 1 hour:** Shipping database logs to the remote site every hour. The corresponding recovery strategy is to recover the database to the point of the last log shipment.
- **RPO in the order of minutes:** Mirroring data asynchronously to a remote site
- **Near zero RPO:** Mirroring data synchronously to a remote site

- **Recovery-Time Objective (RTO):** The time within which systems and applications must be recovered after an outage. It defines the amount of downtime that a business can endure and survive. Businesses can optimize disaster recovery plans after defining the RTO for a given system.

  For example, if the RTO is 2 hours, it requires disk-based backup because it enables a faster restore than a tape backup. However, for an RTO of 1 week, tape backup will likely meet the requirements. Some examples of RTOs and the recovery strategies to ensure data availability are listed here (refer to Figure 9-3 [b]):

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



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

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

- **Data vault:** A repository at a remote site where data can be periodically or continuously copied (either to tape drives or disks) so that there is always a copy at another site

- **Hot site:** A site where an enterprise's operations can be moved in the event of disaster. It is a site with the required hardware, operating system, application, and network support to perform business operations, where the equipment is available and running at all times.
- **Cold site:** A site where an enterprise's operations can be moved in the event of disaster, with minimum IT infrastructure and environmental facilities in place, but not activated
- **Server Clustering:** A group of servers and other necessary resources coupled to operate as a single system. Clusters can ensure high availability and load balancing. Typically, in failover clusters, one server runs an application and updates the data, and another server is kept as standby to take over completely, as required. In more sophisticated clusters, multiple servers may access data, and typically one server is kept as standby.

  Server clustering provides load balancing by distributing the application load evenly among multiple servers within the cluster.

BC Planning Life Cycle:

The BC planning life cycle includes five stages:

1. Establishing objectives

2. Analyzing

3. Designing and developing

4. Implementing

5. Training, testing, assessing, and maintaining

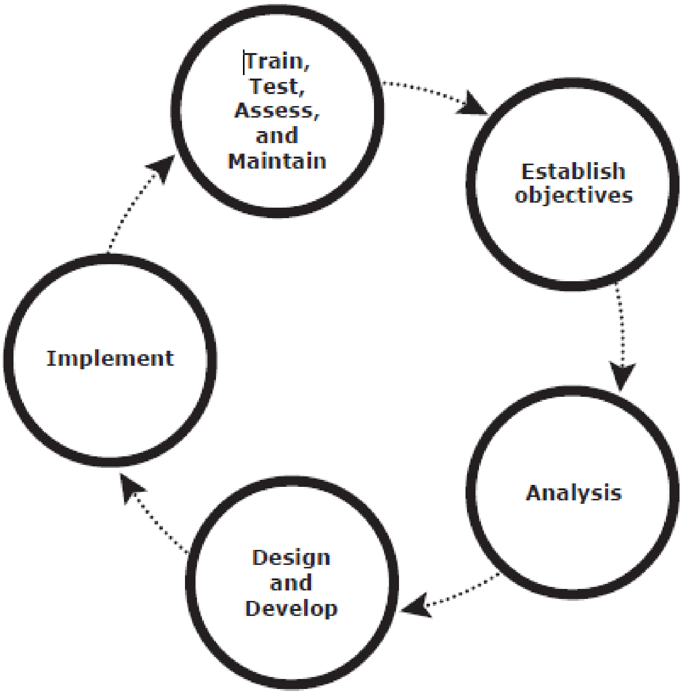


Figure 9-4: BC planning life cycle

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

1. Establish objectives:

- Determine BC requirements.
- Estimate the scope and budget to achieve requirements.

- Select a BC team that includes subject matter experts from all areas of the business, whether internal or external.
- Create BC policies.

2. Analysis:

- Collect information on data profiles, business processes, infrastructure support, dependencies, and frequency of using business infrastructure.
- Conduct a Business Impact Analysis (BIA).
- Identify critical business processes and assign recovery priorities.
- Perform risk analysis for critical functions and create mitigation strategies.
- Perform cost benefit analysis for available solutions based on the mitigation strategy.
- Evaluate options.

3. Design and develop:

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

4. Implement:

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

5. Train, test, assess, and maintain:

- Train the employees who are responsible for backup and replication of business-critical data on a regular basis or whenever there is a modification in the BC plan.
- Train employees on emergency response procedures when disasters are declared.
- Train the recovery team on recovery procedures based on contingency scenarios.

- Perform damage-assessment processes and review recovery plans.
- Test the BC plan regularly to evaluate its performance and identify its limitations.
- Assess the performance reports and identify limitations.
- Update the BC plans and recovery/restart procedures to reflect regular changes within the data center.

Failure Analysis:

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

Single Point of Failure:

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

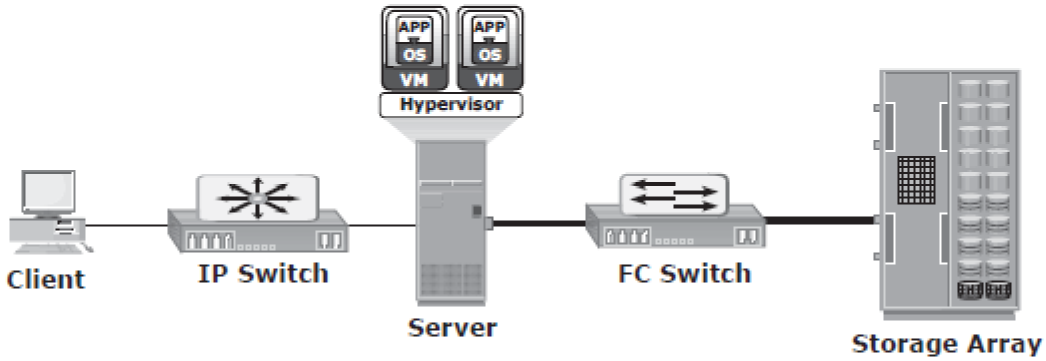


Figure 9-5: Single point of failure

In a setup in which each component must function as required to ensure data availability, the failure of a single physical or virtual component causes the unavailability of an application. This failure results in disruption of business operations.

For example, failure of a hypervisor can affect all the running VMs and the virtual network, which are hosted on it. In the setup shown in Figure 9-5, several single points of failure can be identified.

A VM, a hypervisor, an HBA/NIC on the server, the physical server, the IP network, the FC switch, the storage array ports, or even the storage array could be a potential single point of failure.

Resolving Single Points of Failure:

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

Data centers follow stringent guidelines to implement fault tolerance for uninterrupted information availability. Careful analysis is performed to eliminate every single point of failure. The example shown in Figure 9-6 represents all enhancements in the infrastructure to mitigate single points of failure:

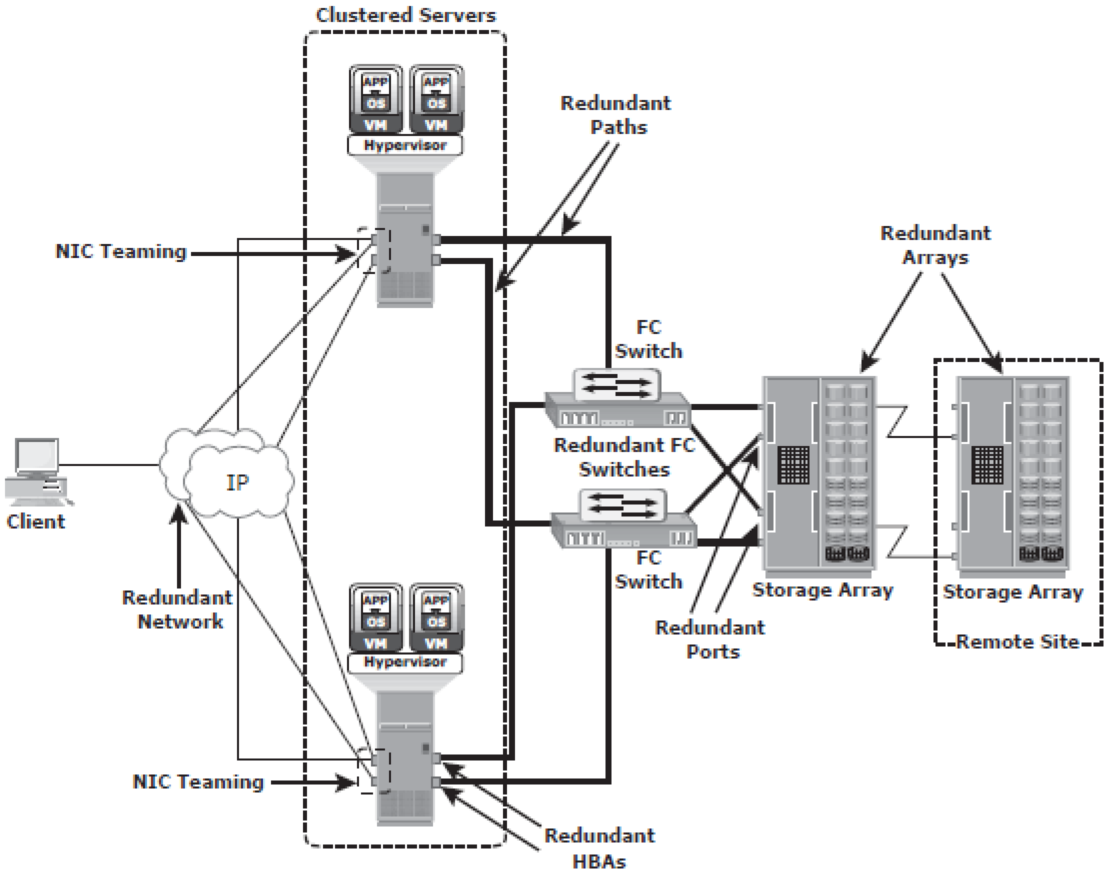


Figure 9-6: Resolving single points of failure

- Configuration of redundant HBAs at a server to mitigate single HBA failure
- Configuration of NIC teaming at a server allows protection against single physical NIC failure. It allows grouping of two or more physical NICs and treating them as a single logical device. With NIC teaming, if one of the underlying physical NICs fails or its cable is unplugged, the traffic is redirected to another physical NIC in the team. Thus, NIC teaming eliminates the single point of failure associated with a single physical NIC.
- Configuration of redundant switches to account for a switch failure
- Configuration of multiple storage array ports to mitigate a port failure
- RAID and hot spare configuration to ensure continuous operation in the event of disk failure
- Implementation of a redundant storage array at a remote site to mitigate local site failure
- Implementing server (or compute) clustering, a fault-tolerance mechanism whereby two or more servers in a cluster access the same set of data volumes. Clustered servers

exchange a heartbeat to inform each other about their health. If one of the servers or hypervisors fails, the other server or hypervisor can take up the workload.

- Implementing a VM Fault Tolerance mechanism ensures BC in the event of a server failure. This technique creates duplicate copies of each VM on another server so that when a VM failure is detected, the duplicate VM can be used for failover. The two VMs are kept in synchronization with each other in order to perform successful failover.

Multipathing Software

Configuration of multiple paths increases the data availability through path failover. If servers are configured with one I/O path to the data, there will be no access to the data if that path fails.

Redundant paths to the data eliminate the possibility of the path becoming a single point of failure. Multiple paths to data also improve I/O performance through load balancing among the paths and maximize server, storage, and data path utilization. In practice, merely configuring multiple paths does not serve the purpose. Even with multiple paths, if one path fails, I/O does not reroute unless the system recognizes that it has an alternative path.

Multipathing software provides the functionality to recognize and utilize alternative I/O paths to data. Multipathing software also manages the load balancing by distributing I/Os to all available, active paths.

Multipathing software intelligently manages the paths to a device by sending I/O down the optimal path based on the load balancing and failover policy setting for the device. It also takes into account path usage and availability before deciding the path through which to send the I/O. If a path to the device fails, it automatically reroutes the I/O to an alternative path.

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

Business Impact Analysis:

A business impact analysis (BIA) identifies which business units, operations, and processes are essential to the survival of the business. It evaluates the financial, operational, and service impacts of a disruption to essential business processes.

Selected functional areas are evaluated to determine resilience of the infrastructure to support information availability.

The BIA process leads to a report detailing the incidents and their impact over business functions. The impact may be specified in terms of money or in terms of time. Based on the potential impacts associated with downtime, businesses can prioritize and implement countermeasures to mitigate the likelihood of such disruptions. These are detailed in the BC plan.

A BIA includes the following set of tasks:

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

BC Technology Solutions

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

Backup: Data backup is a predominant method of ensuring data availability. The frequency of backup is determined based on RPO, RTO, and the frequency of data changes.

Local replication: Data can be replicated to a separate location within the same storage array. The replica is used independently for other business operations. Replicas can also be used for restoring operations if data corruption occurs.

Remote replication: Data in a storage array can be replicated to another storage array located at a remote site. If the storage array is lost due to a disaster, business operations can be started from the remote storage array.

Backup and Archive

A **backup** is an additional copy of production data, created and retained for the sole purpose of recovering lost or corrupted data.

Data archiving is the process of moving data that is no longer actively used, from primary storage to a low-cost secondary storage. The data is retained in the secondary storage for a long term to meet regulatory requirements.

Backup Methods:

Hot backup and **cold backup** are the two methods deployed for a backup. They are based on the state of the application when the backup is performed.

In a hot backup, the application is up-and-running, with users accessing their data during the backup process. This method of backup is also referred to as an online backup.

A cold backup requires the application to be shut down during the backup process. Hence, this method is also referred to as an offline backup.

The hot backup of online production data is challenging because data is actively used and changed. If a file is open, it is normally not backed up during the backup process. In such situations, an open file agent is required to back up the open file. These agents interact directly with the operating system or application and enable the creation of consistent copies of open files.

In database environments, the use of open file agents is not enough, because the agent should also support a consistent backup of all the database components. For example, a database is composed of many files of varying sizes occupying several file systems. To ensure a consistent database backup, all files need to be backed up in the same state. That does not necessarily mean that all files need to be backed up at the same time, but they all must be synchronized so that the database can be restored with consistency.

The disadvantage associated with a hot backup is that the agents usually affect the overall application performance.

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

A point-in-time (PIT) copy method is deployed in environments in which the impact of downtime from a cold backup or the performance impact resulting from a hot backup is unacceptable.

The PIT copy is created from the production volume and used as the source for the backup. This reduces the impact on the production volume.

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

In a disaster recovery environment, bare-metal recovery (BMR) refers to a backup in which all metadata, system information, and application configurations are appropriately backed up for a full system recovery. BMR builds the base system, which includes partitioning, the file system layout, the operating system, the applications, and all the relevant configurations.

BMR recovers the base system first before starting the recovery of data files. Some BMR technologies — for example server configuration backup (SCB) — can recover a server even onto dissimilar hardware.

Backup Topologies:

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

Direct-attached backup:

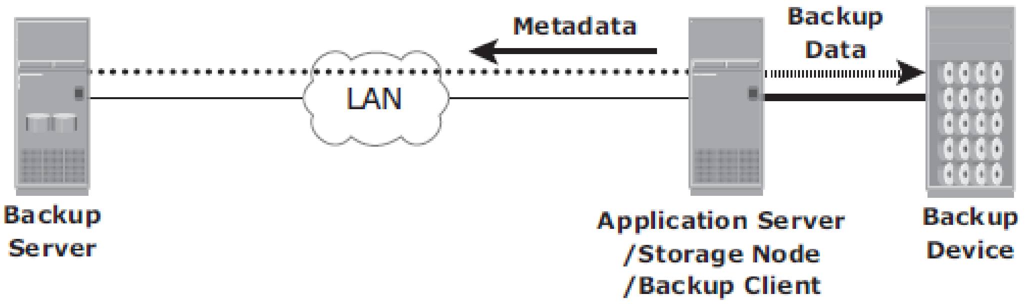


Figure 10-7: Direct-attached backup topology

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

LAN-based backup:

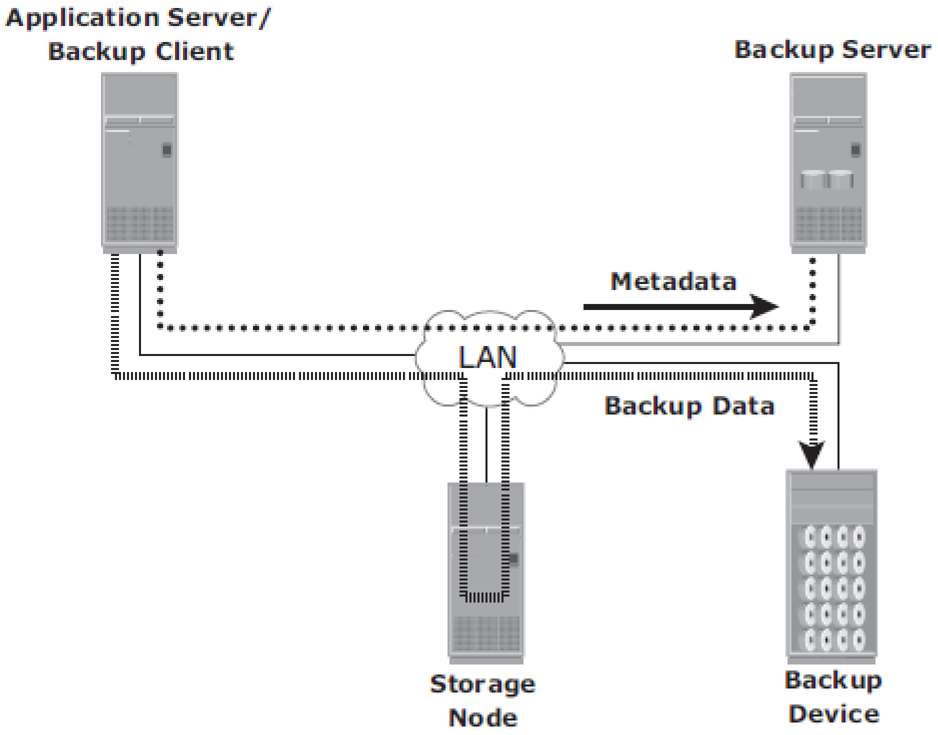


Figure 10-8: LAN-based backup topology

- In a LAN-based backup, the clients, backup server, storage node, and backup device are connected to the LAN. (see Figure 10-8).
- The data to be backed up is transferred from the backup client (source) to the backup device (destination) over the LAN, which might affect network performance.

- 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.

SAN-based backup:

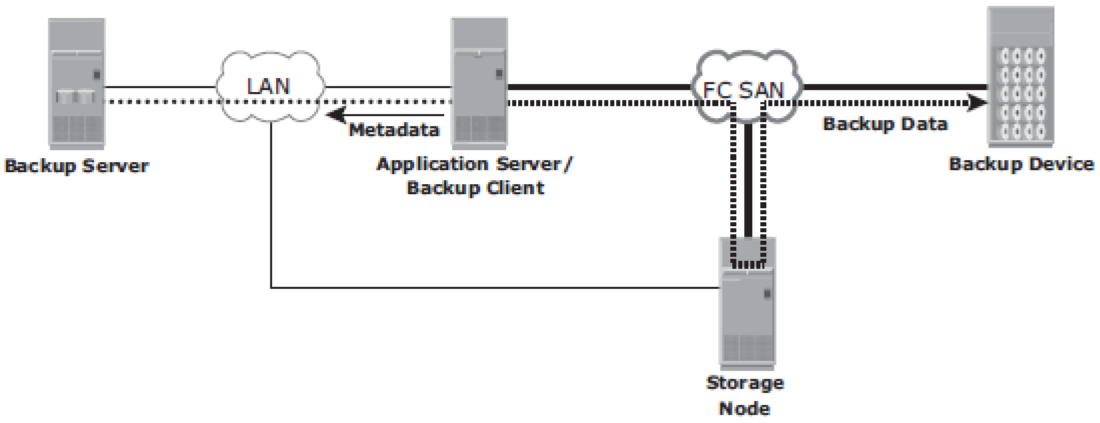


Figure 10-9: SAN-based backup topology

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

Figure 10-9 illustrates a SAN-based backup. In this example, a client sends the data to be backed up to the backup device over the SAN. Therefore, the backup data traffic is restricted to the SAN, and only the backup metadata is transported over the LAN. The volume of metadata is insignificant when compared to the production data; the LAN performance is not degraded in this configuration.

Mixed Topology:

The mixed topology uses both the LAN-based and SAN-based topologies, as shown in Figure 10-10.

This topology might be implemented for several reasons, including cost, server location, reduction in administrative overhead, and performance considerations.

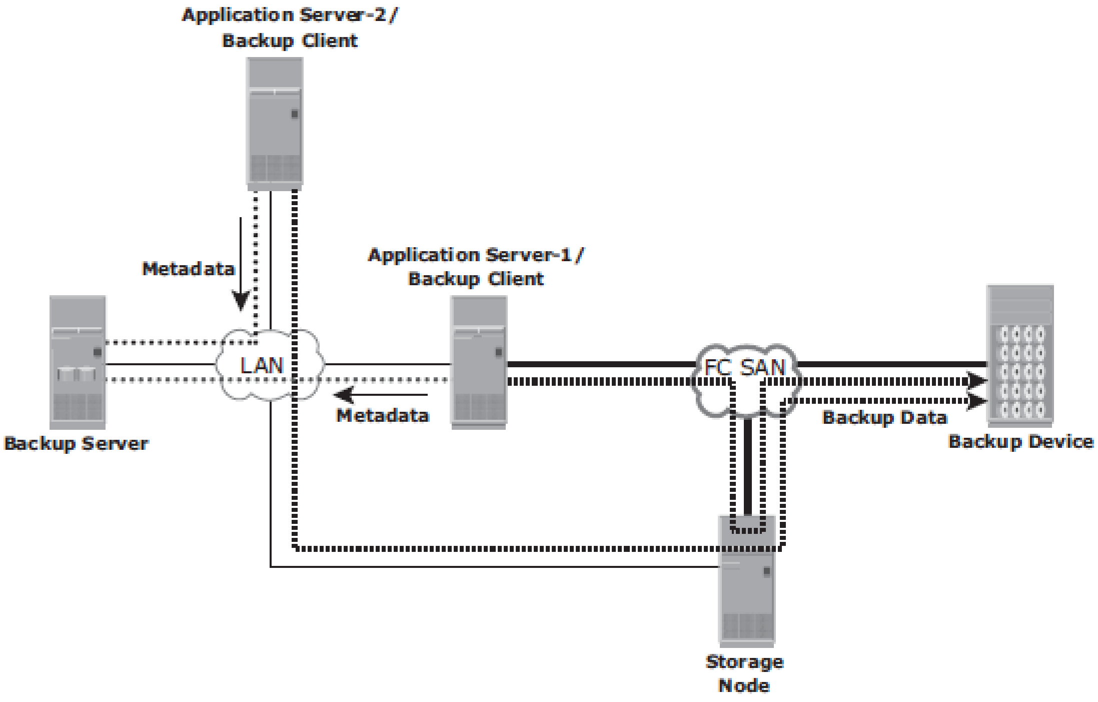


Figure 10-10: Mixed backup topology

Backup Targets:

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

Backup to Tape:

Tapes, a low-cost solution, are used extensively for backup. Tape drives are used to read/write data from/to a tape cartridge (or cassette). 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, density, tape length, tape thickness, tape tracks, and supported speed.

Physical Tape Library

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

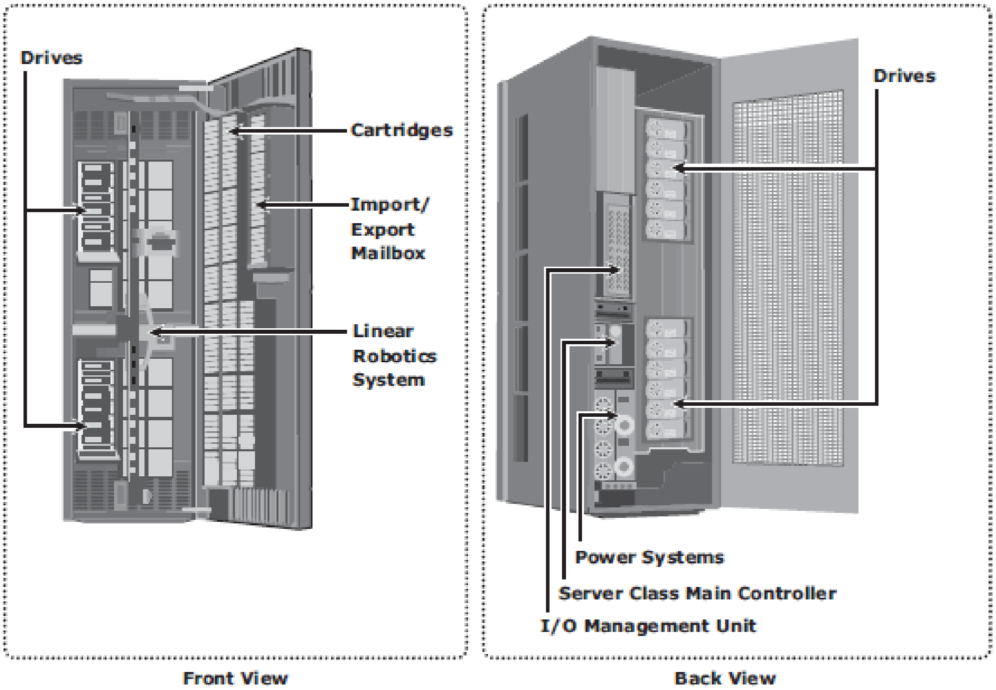


Figure 10-15: Physical tape library

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

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

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

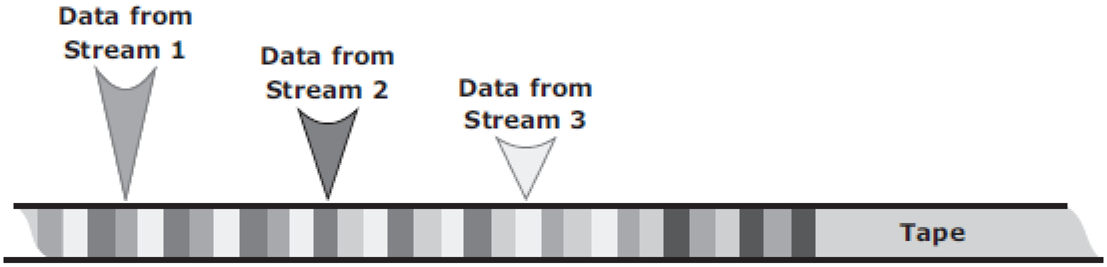


Figure 10-16: Multiple streams on tape media

Many times, even the buffering and speed adjustment features of a tape drive fail to prevent the gaps, causing the “shoe shining effect” or “backhitching.” Shoe shining is the repeated back and forth motion a tape drive makes when there is an interruption in the backup data stream.

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

When a restore is initiated, the backup software identifies which tapes are required. The robotic arm is instructed to move the tape from its slot to a tape drive. If the required tape is not found in the tape library, the backup software displays a message, instructing the operator to manually insert the required tape in the tape library.

When a file or a group of files require restores, the tape must move to that file location sequentially before it can start reading. This process can take a significant amount of time, especially if the required files are recorded at the end of the tape.

Limitations of Tape:

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

Backup to Disk

Because of increased availability, low cost disks have now replaced tapes as the primary device for storing backup data because of their performance advantages.

Backup-to-disk systems offer ease of implementation, reduced TCO, and improved quality of service. Apart from performance benefits in terms of data transfer rates, disks also offer faster recovery when compared to tapes.

Backing up to disk storage systems offers clear advantages due to their inherent random access and RAID-protection capabilities. In most backup environments, backup to disk is used as a staging area where the data is copied temporarily before transferring or staging it to tapes. This enhances backup performance.

Some backup products allow for backup images to remain on the disk for a period of time even after they have been staged. This enables a much faster restore.

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

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

Backup to Virtual Tape:

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

Virtual Tape Library

A virtual tape library (VTL) has the same components as that of a physical tape library, except that the majority of the components are presented as virtual resources. For the backup software, there is no difference between a physical tape library and a virtual tape library. Figure 10-18 shows a virtual tape library.

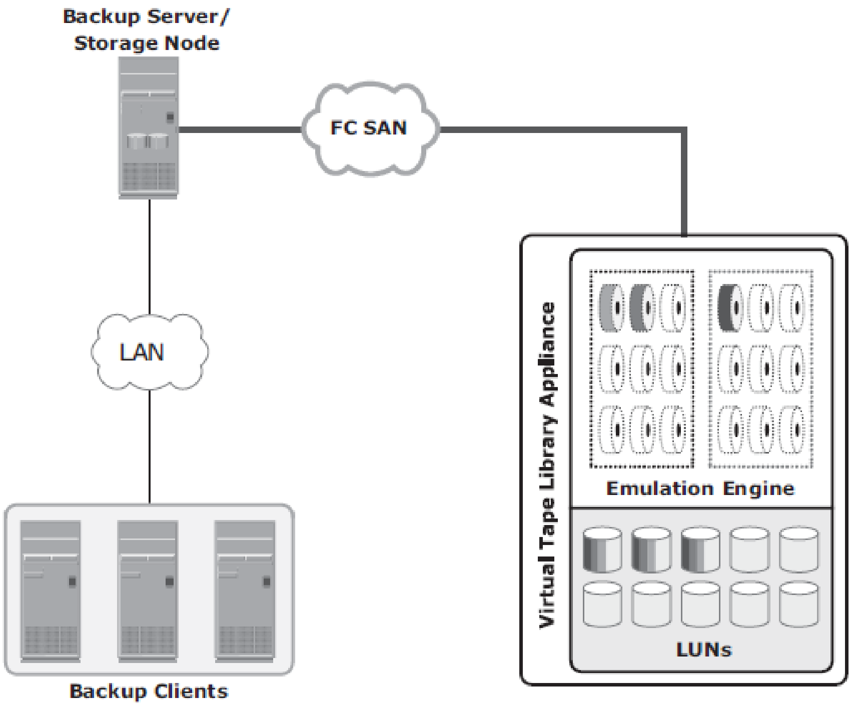


Figure 10-18: Virtual tape library

Virtual tape libraries use disks as backup media. Emulation software has a database with a list of virtual tapes, and each virtual tape is assigned space on a LUN. A virtual tape can span multiple LUNs if required. File system awareness is not required while backing up because the virtual tape solution typically uses raw devices.

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

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

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

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

Data replication over IP is available with most of the virtual tape library appliances. This feature enables virtual tapes to be replicated over an inexpensive IP network to a remote site. As a result, organizations can comply with offsite requirements for backup data. Connecting the engines of a virtual tape library appliance to a physical tape library enables the virtual tapes to be copied onto the physical tapes, which can then be sent to a vault or shipped to an offsite location.

Using virtual tapes offers several advantages over both physical tapes and disks. Compared to physical tapes, virtual tapes offer better single stream performance, better reliability, and random disk access characteristics. Backup and restore operations benefit from the disk's random access characteristics because they are always online and provide faster backup and recovery.

A virtual tape drive does not require the usual maintenance tasks associated with a physical tape drive, such as periodic cleaning and drive calibration. Compared to backup-to-disk devices, a virtual tape library offers easy installation and administration because it is preconfigured by the manufacturer.

However, a virtual tape library is generally used only for backup purposes. In a backup-to-disk environment, the disk systems are used for both production and backup data.

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.

Data Deduplication Methods:

There are two methods of deduplication: file level and subfile level.

Determining the uniqueness by implementing either method offers benefits; however, results can vary. The differences exist in the amount of data reduction each method produces and the time each approach takes to determine the unique content.

File-level deduplication:

- 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:

- 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 This shifts the whole document, and all the blocks appear to have changed, causing the failure of the deduplication method to detect equivalencies.
- In variable-length segment deduplication, if there is a change in the segment, the boundary for only that segment is adjusted, leaving the remaining segments unchanged. This method vastly improves the ability to find duplicate data segments compared to fixed-block.

Data Deduplication Implementation

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

Source-Based Data Deduplication:

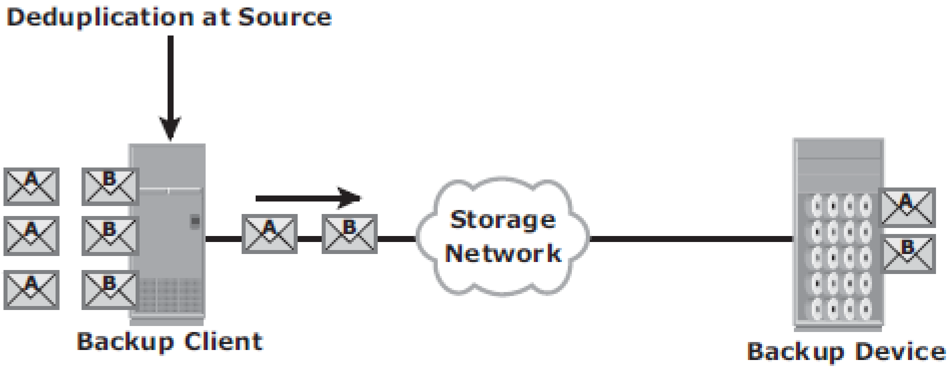


Figure 10-19: Source-based data deduplication

- Source-based data deduplication eliminates redundant data at the source before it transmits to the backup device.
- Source-based data deduplication can dramatically reduce the amount of backup data sent over the network during backup processes.
- It provides the benefits of a shorter backup window and requires less network bandwidth.
- There is also a substantial reduction in the capacity required to store the backup images. Figure 10-19 shows source-based data deduplication.
- Source-based deduplication increases the overhead on the backup client, which impacts the performance of the backup and application running on the client.
- Source-based deduplication might also require a change of backup software if it is not supported by backup software.

Target-Based Data Deduplication:

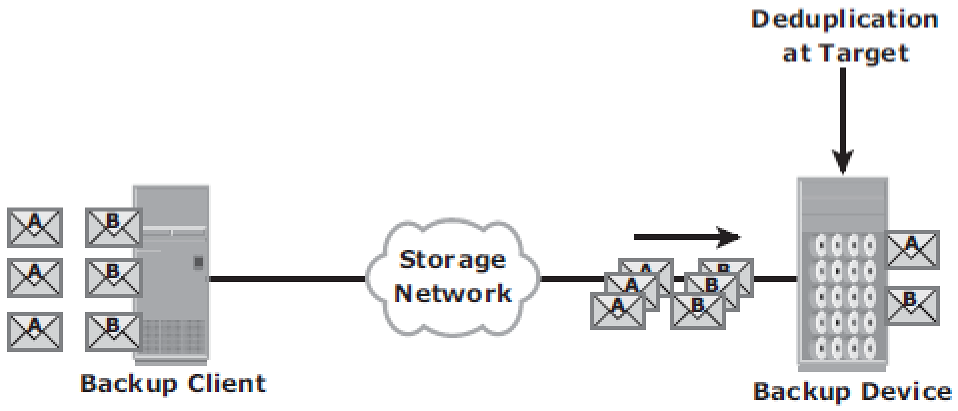


Figure 10-20: Target-based data deduplication

- Target-based data deduplication is an alternative to source-based data deduplication
- Target-based data deduplication occurs at the backup device, which offloads the backup client from the deduplication process.
- Figure 10-20 shows target-based data deduplication.
- In this case, the backup client sends the data to the backup device and the data is deduplicated at the backup device, either immediately (inline) or at a scheduled time (post-process). Because deduplication occurs at the target, all the backup data needs to be transferred over the network, which increases network bandwidth requirements.
- Target-based data deduplication does not require any changes in the existing backup software.
- **Inline deduplication :**

  - Inline deduplication performs deduplication on the backup data before it is stored on the backup device. Hence, this method reduces the storage capacity needed for the backup.
  - Inline deduplication introduces overhead in the form of the time required to identify and remove duplication in the data. So, this method is best suited for an environment with a large backup window.
- **Post-process deduplication:**
  - Post-process deduplication enables the backup data to be stored or written on the backup device first and then deduplicated later. This method is suitable for situations with tighter backup windows. However, post-process deduplication requires more storage capacity to store the backup images before they are deduplicated.

Backup in Virtualized Environments:

In a virtualized environment, it is imperative to back up the virtual machine data (OS, application data, and configuration) to prevent its loss or corruption due to human or technical errors.

There are two approaches for performing a backup in a virtualized environment:

i. The traditional backup approach and
ii. The image-based backup approach.

Traditional backup approach:

In the traditional backup approach, a backup agent is installed either on the virtual machine (VM) or on the hypervisor. Figure 10-21 shows the traditional VM backup approach.

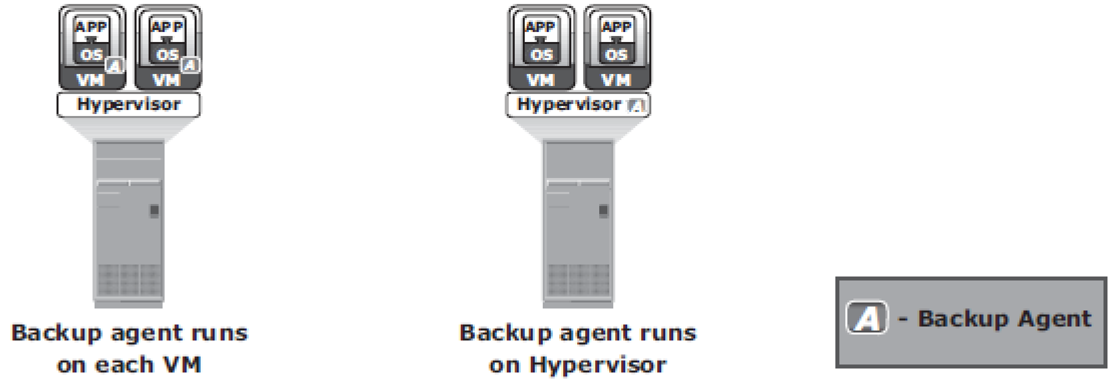


Figure 10-21: Traditional VM backup

If the backup agent is installed on a VM, the VM appearsas a physical server to the agent. The backup agent installed on the VM backs up the VM data to the backup device. The agent does not capture VM files, such as the virtual BIOS file, VM swap file, logs, and configuration files. Therefore, for a VM restore, a user needs to manually re-create the VM and then restore data onto it.

If the backup agent is installed on the hypervisor, the VMs appear as a set of files to the agent. So, VM files can be backed up by performing a filesystem 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.

Image-based backup approach:

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

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

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.

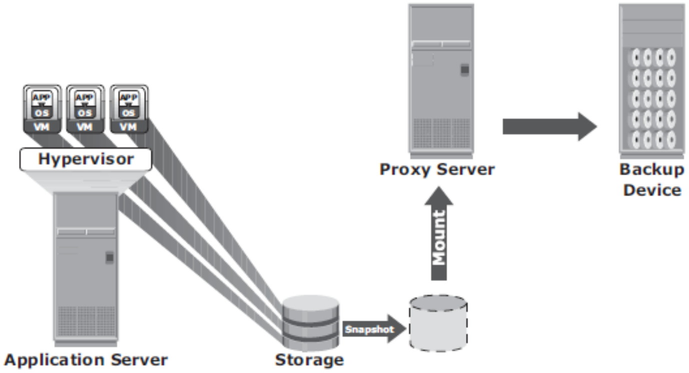
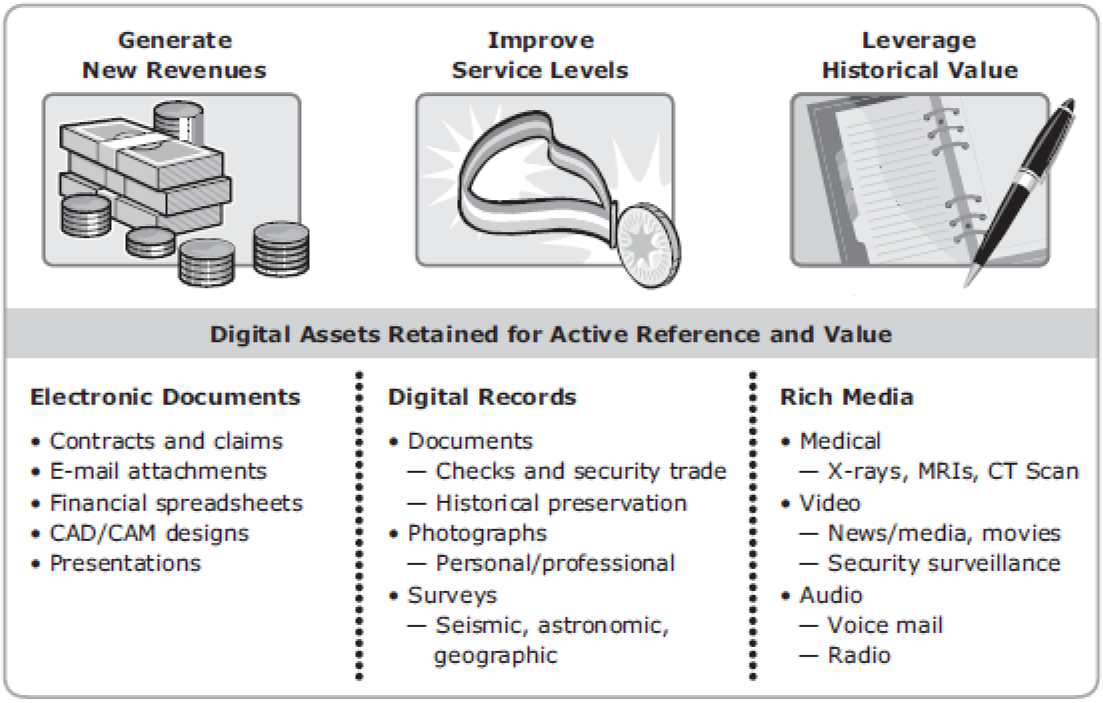


Figure 10-22: Image-based backup

The backup software then backs up these image files normally. (see Figure 10-22). This effectively offloads the backup processing from the hypervisor and transfers the load on the proxy server, thereby reducing the impact to VMs running on the hypervisor. Image-based backup enables quick restoration of a VM.

Data Archive:

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



Generate New Revenues
Improve Service Levels
Leverage Historical Value
Digital Assets Retained for Active Reference and Value

Electronic Documents

- Contracts and claims
- E-mail attachments
- Financial spreadsheets
- CAD/CAM designs
- Presentations

Digital Records

- Documents
  - — Checks and security trade
  - — Historical preservation
- Photographs
  - — Personal/professional
- Surveys
  - — Seismic, astronomic, geographic

Rich Media

- Medical
  - — X-rays, MRIs, CT Scan
- Video
  - — News/media, movies
  - — Security surveillance
- Audio
  - — Voice mail
  - — Radio

Figure 10-23: Examples of fixed content data

All organizations may require retention of their data for an extended period of time due to government regulations and legal/contractual obligations.

Organizations also make use of this fixed content to generate new revenue strategies and improve service levels. A repository where fixed content is stored is known as an **archive**.

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

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

Traditionally, optical and tape media were used for archives. Optical media are typically write once read many (WORM) devices that protect the original file from being overwritten. Some tape devices also provide this functionality by implementing file-locking capabilities. Although these devices are inexpensive, they involve operational, management, and maintenance overhead. The traditional archival process using optical discs and tapes is not optimized to recognize the content, so the same content could be archived several times. Additional costs are involved in offsite storage of media and media management.

Tapes and optical media are also susceptible to wear and tear. Frequent changes in these device technologies lead to the overhead of converting the media into new formats to enable access and retrieval. Government agencies and industry regulators are establishing new laws and regulations to enforce the protection of archives from unauthorized destruction and modification. These regulations and standards have established new requirements for preserving the integrity of information in the archives. These requirements have exposed the shortcomings of the traditional tape and optical media archive solutions.

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

Local Replication

Replication is one of the ways to ensure BC. It is the process to create an exact copy (replica) of data. These replica copies are used for restore and restart operations if data loss occurs.

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

Replication Terminology:

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

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

Uses of Local Replicas:

- **Alternative source for backup:** Under normal backup operations, data is read from the production volumes (LUNs) and written to the backup device. This places an additional burden on the production infrastructure because production LUNs are simultaneously involved in production. operations and servicing data for backup operations. The local replica contains an exact point-in-time (PIT) copy of the source data, and therefore can be used as a source to perform backup operations. This alleviates the backup I/O workload on the production volumes. Another benefit of using local replicas for backup is that it reduces the backup window to zero.
- **Fast recovery:** If data loss or data corruption occurs on the source, a local replica might be used to recover the lost or corrupted data. If a complete failure of the source occurs, some replication solutions enable a replica to be used to restore data onto a different set of source devices, or production can be restarted on the replica. In either case, this method provides faster recovery and minimal RTO compared to traditional recovery from tape backups. In many instances, business operations can be started using the source device before the data is completely copied from the replica.
- **Decision-support activities, such as reporting or data warehousing:** Running the reports using the data on the replicas greatly reduces the I/O burden placed on the production device. Local replicas are also used for data-warehousing applications. The data-warehouse application may be populated by the data on the replica and thus avoid the impact on the production environment.
- **Testing platform:** Local replicas are also used for testing new applications or upgrades. For example, an organization may use the replica to test the production application upgrade; if the test is successful, the upgrade may be implemented on the production environment.
- **Data migration:** Another use for a local replica is data migration. Data migrations are performed for various reasons, such as migrating from a smaller capacity LUN to one of a larger capacity for newer versions of the application.

Local Replication Technologies:

Host-based, storage array-based, and network-based replications are the major technologies used for local replication.

Host-Based Local Replication:

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

LVM-Based Replication:

- In LVM-based replication, the logical volume manager is responsible for creating and controlling the host-level logical volumes.
- An LVM has three components: physical volumes (physical disk), volume groups, and logical volumes.
- A volume group is created by grouping one or more physical volumes.
- Logical volumes are created within a given volume group.
- A volume group can have multiple logical volumes.
- In LVM-based replication, each logical block in a logical volume is mapped to two physical blocks on two different physical volumes, as shown in Figure 11-5.
- An application write to a logical volume is written to the two physical volumes by the LVM device driver. This is also known as LVM mirroring. Mirrors can be split, and the data contained therein can be independently accessed.

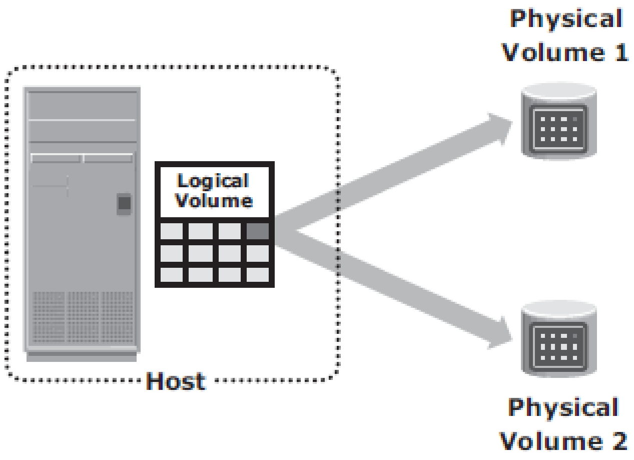


Figure 11-5: LVM-based mirroring

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:

- Every write generated by an application translates into two writes on the disk, and thus, an additional burden is placed on the host CPU. This can degrade application performance.
- Presenting an LVM-based local replica to another host is usually not possible because the replica will still be part of the volume group, which is usually accessed by one host at any given time.
- Tracking changes to the mirrors and performing incremental resynchronization operations is also a challenge because all LVMs do not support incremental resynchronization.
- If the devices are already protected by some level of RAID on the array, then the additional protection that the LVM mirroring provides is unnecessary. This solution does not scale to provide replicas of federated databases and applications.
- Both the replica and source are stored within the same volume group. Therefore, the replica might become unavailable if there is an error in the volume group. If the server fails, both the source and replica are unavailable until the server is brought back online.

File System Snapshot

- A file system (FS) snapshot is a pointer-based replica that requires a fraction of the space used by the production FS. This snapshot can be implemented by either FS or by LVM. It uses the Copy on First Write (CoFW) principle to create snapshots.
- When a snapshot is created, a bitmap and blockmap are created in the metadata of the Snap FS.
- The bitmap is used to keep track of blocks that are changed on the production FS after the snap creation.
- The blockmap is used to indicate the exact address from which the data is to be read when the data is accessed from the Snap FS.
- Immediately after the creation of the FS snapshot, all reads from the snapshot are actually served by reading the production FS.
- In a CoFW mechanism, if a write I/O is issued to the production FS for the first time after the creation of a snapshot, the I/O is held and the original data of production FS

corresponding to that location is moved to the Snap FS. Then, the write is allowed to the production FS.

- The bitmap and blockmap are updated accordingly. Subsequent writes to the same location do not initiate the CoFW activity.
- To read from the Snap FS, the bitmap is consulted. If the bit is 0, then the read is directed to the production FS. If the bit is 1, then the block address is obtained from the blockmap, and the data is read from that address on the Snap FS.
- Read requests from the productionFS work as normal.

Figure 11-6 illustrates the write operations to the production file system.

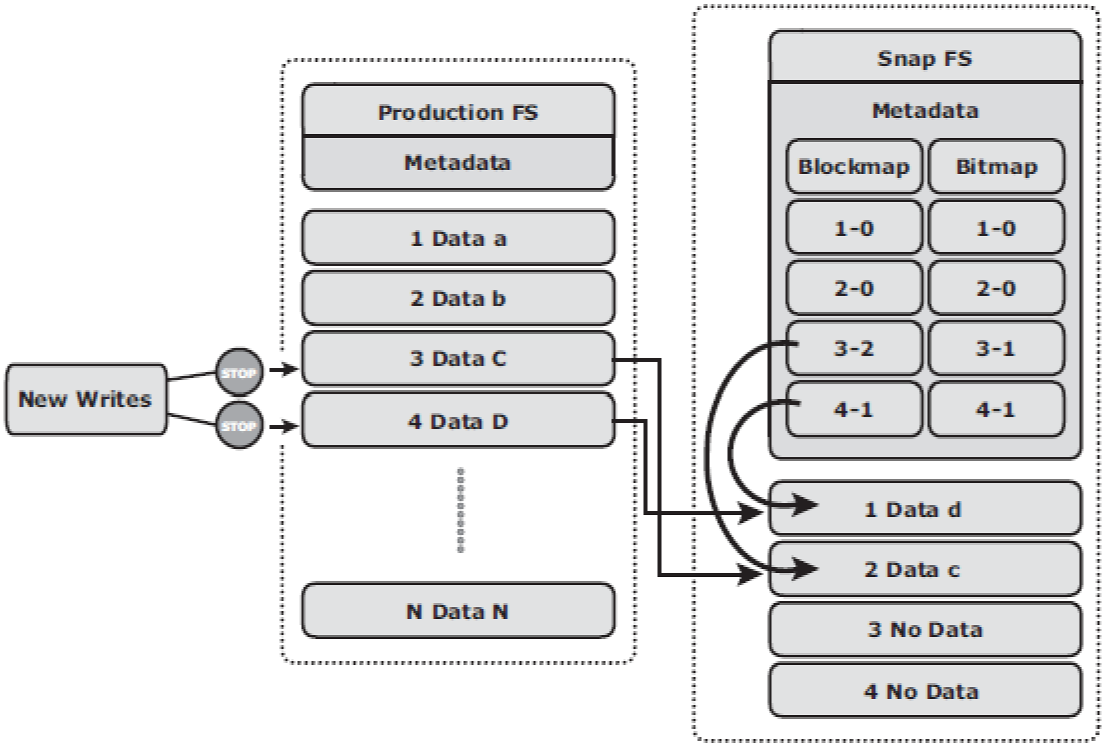


Figure 11-6: Write to production FS

For example, a write data “C” occurs on block 3 at the production FS, which currently holds data “c”' The snapshot application holds the I/O to the production FS and first copies the old data “c” to an available data block on the Snap FS. The bitmap and blockmap values for block 3 in the production FS are changed in the snap metadata. The bitmap of block 3 is changed to 1, indicating that this block has changed on the production FS. The block map of block 3 is changed and indicates the block number where the data is written in Snap FS, (in this case block

2). After this is done, the I/Os to the production FS are allowed to complete. Any subsequent writes to block3 on the production FS occur as normal, and it does not initiate the CoFW operation.

Similarly, if an I/O is issued to block 4 on the production FS to change the value of data “d” to “D,” the snapshot application holds the I/O to the production FS and copies the old data to an available data block on the Snap FS. Then it changes the bitmap of block 4 to 1, indicating that the data block has changed on the production FS. The blockmap for block 4 indicates the block number where the data can be found on the Snap FS, in this case, data block 1 of the Snap FS. After this is done, the I/O to the production FS is allowed to complete.

Storage Array-Based Local Replication:

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

In this replication, the required number of replica devices should be selected on the same array and then data should be replicated between the source-replica pairs.

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

Full-Volume Mirroring:

- In full-volume mirroring, the target is attached to the source and established as a mirror of the source (Figure 11-8 [a]).
- The data on the source is copied to the target. New updates to the source are also updated on the target.
- After all the data is copied and both the source and the target contain identical data, the target can be considered as a mirror of the source.
- While the target is attached to the source it remains unavailable to any other host. However, the production host continues to access the source.
- After the synchronization is complete, the target can be detached from the source and made available for other business operations.
- Figure 11-8 (b) shows full-volume mirroring when the target is detached from the source.

- Both the source and the target can be accessed for read and write operations by the production and business continuity hosts respectively.
- After detaching from the source, the target becomes a point-in-time (PIT) copy of the source. The PIT of a replica is determined by the time when the target is detached from the source. For example, if the time of detachment is 4:00 p.m., the PIT for the target is 4:00 p.m. After detachment, changes made to both the source and replica can be tracked at some predefined granularity. This enables incremental resynchronization (source to target) or incremental restore (target to source). The granularity of the data change can range from 512 byte blocks to 64 KB blocks or higher.

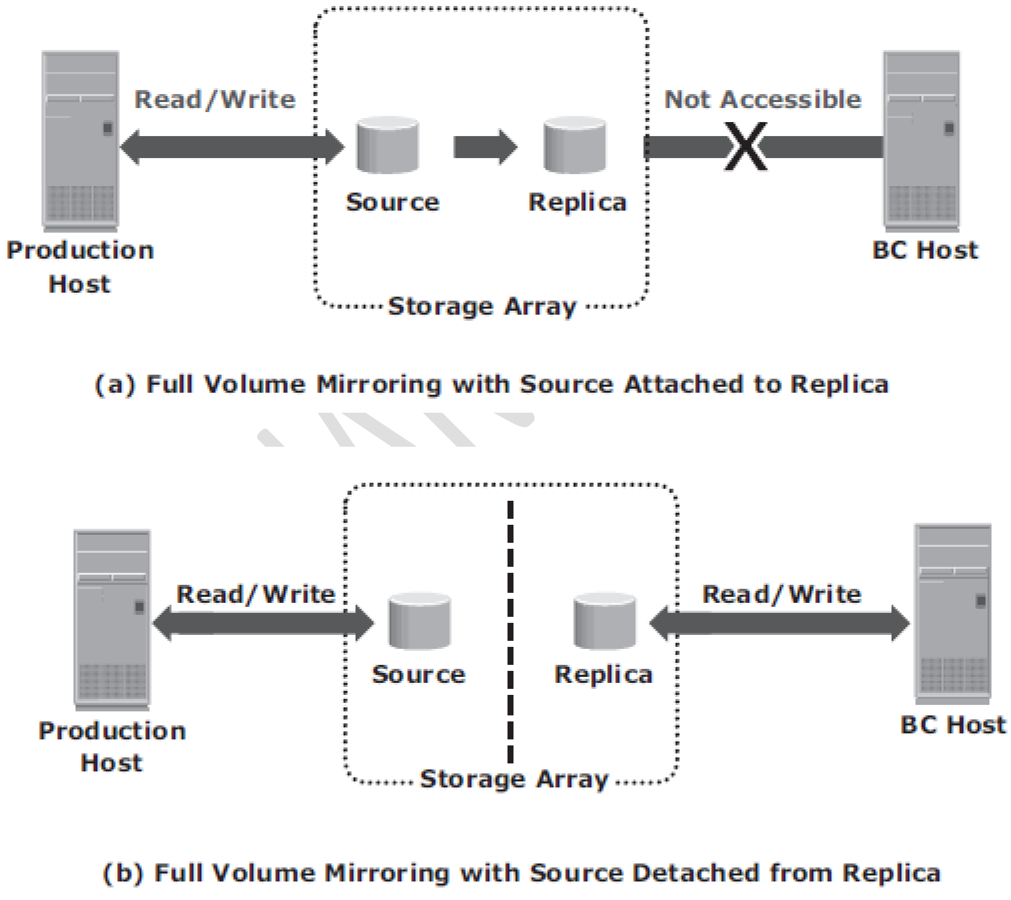


Figure 11-8: Full-volume mirroring

Pointer-Based, Full-Volume Replication:

- In Pointer-Based, Full-Volume Replication the target is immediately accessible by the BC host after the replication session is activated.
- Pointer-based, full-volume replication can be activated in either Copy on First Access (CoFA) mode or Full Copy mode. In either case, at the time of activation, a protection bitmap is created for all data on the source devices.
- The protection bitmap keeps track of the changes at the source device.
- The pointers on the target are initialized to map the corresponding data blocks on the source.
- The data is then copied from the source to the target based on the mode of activation.
- In CoFA, after the replication session is initiated, the data is copied from the source to the target only when the following condition occurs:
  - A write I/O is issued to a specific address on the source for the first time.
  - A read or write I/O is issued to a specific address on the target for the first time.
- When a write is issued to the source for the first time after replication session activation, the original data at that address is copied to the target.
- After this operation, the new data is updated on the source. This ensures that the original data at the point-in-time of activation is preserved on the target (see Figure 11-9).

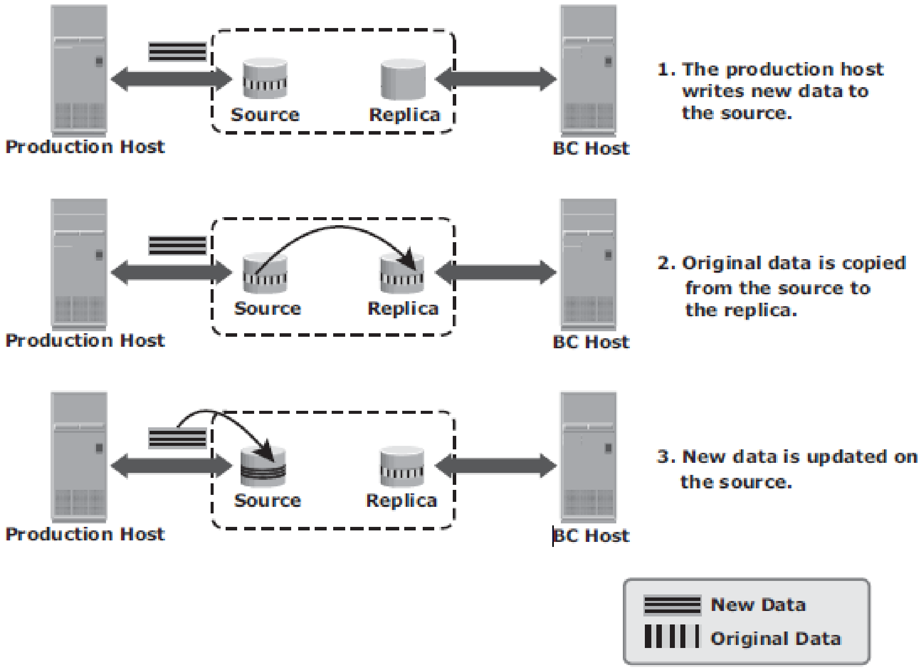

Production Host
Source
Replica
BC Host
1. The production host writes new data to the source.
Production Host
Source
Replica
BC Host
2. Original data is copied from the source to the replica.
Production Host
Source
Replica
BC Host
3. New data is updated on the source.
New Data
Original Data

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

- When a read is issued to the target for the first time after replication session activation, the original data is copied from the source to the target and is made available to the BC host (see Figure 11-10).

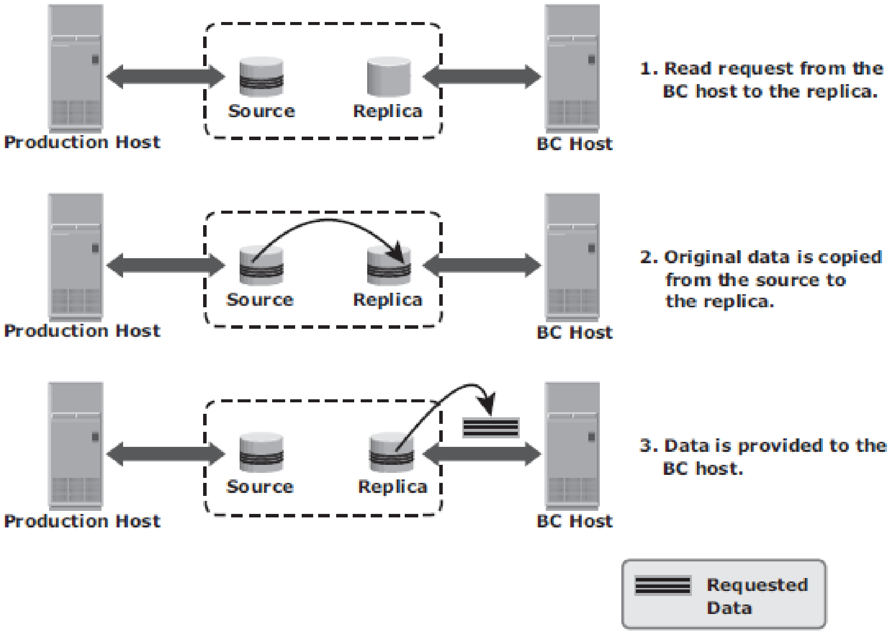

Production Host
Source
Replica
BC Host
1. Read request from the BC host to the replica.
2. Original data is copied from the source to the replica.
3. Data is provided to the BC host.
Requested Data

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

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

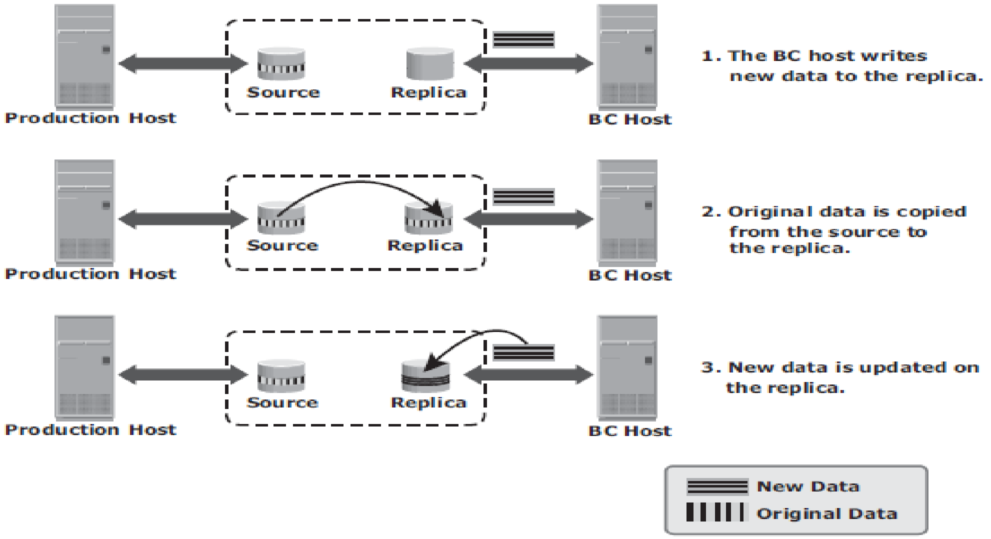

Production Host
Source
Replica
BC Host
1. The BC host writes new data to the replica.
2. Original data is copied from the source to the replica.
3. New data is updated on the replica.
New Data
Original Data

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

- In all cases, the protection bit for the data block on the source is reset to indicate that the original data has been copied over to the target.
- The pointer to the source data can now be discarded. Subsequent writes to the same data block on the source, and the reads or writes to the same data blocks on the target, do not trigger a copy operation, therefore this method is termed “Copy on First Access.”
- If the replication session is terminated, then the target device has only the data that was accessed until the termination, not the entire contents of the source at the point-in-time. In this case, the data on the target cannot be used for restore because it is not a full replica of the source.
- In a Full Copy mode, all data from the source is copied to the target in the background. Data is copied regardless of access.
- If access to a block that has not yet been copied to the target is required, this block is preferentially copied to the target.
- In a complete cycle of the Full Copy mode, all data from the source is copied to the target. If the replication session is terminated now, the target contains all the original data from the source at the point-in-time of activation. This makes the target a viable copy for restore or other business continuity operations.
- The key difference between a pointer-based, Full Copy mode and full-volume mirroring is that the target is immediately accessible upon replication session activation in the Full Copy mode.
- Both the full-volume mirroring and pointer based full-volume replication technologies require the target devices to be at least as large as the source devices.
- The full-volume mirroring and pointer-based full-volume replication in the Full Copy mode can provide incremental resynchronization and restore capabilities.

Pointer-Based Virtual Replication:

In pointer-based virtual replication, at the time of the replication session activation, the target contains pointers to the location of the data on the source. The target does not contain data at any time. Therefore, the target is known as a virtual replica.

Similar to pointer-based full-volume replication, the target is immediately accessible after the replication session activation. A protection bitmap is created for all data blocks on the source device. Granularity of data blocks can range from 512 byte blocks to 64 KB blocks or greater.

Pointer-based virtual replication uses the CoFW technology. When a write is issued to the source for the first time after the replication session activation, the original data at that address is copied to a predefined area in the array. This area is generally known as the save location.

The pointer in the target is updated to point to this data in the save location. After this, the new write is updated on the source. This process is illustrated in Figure 11-12.

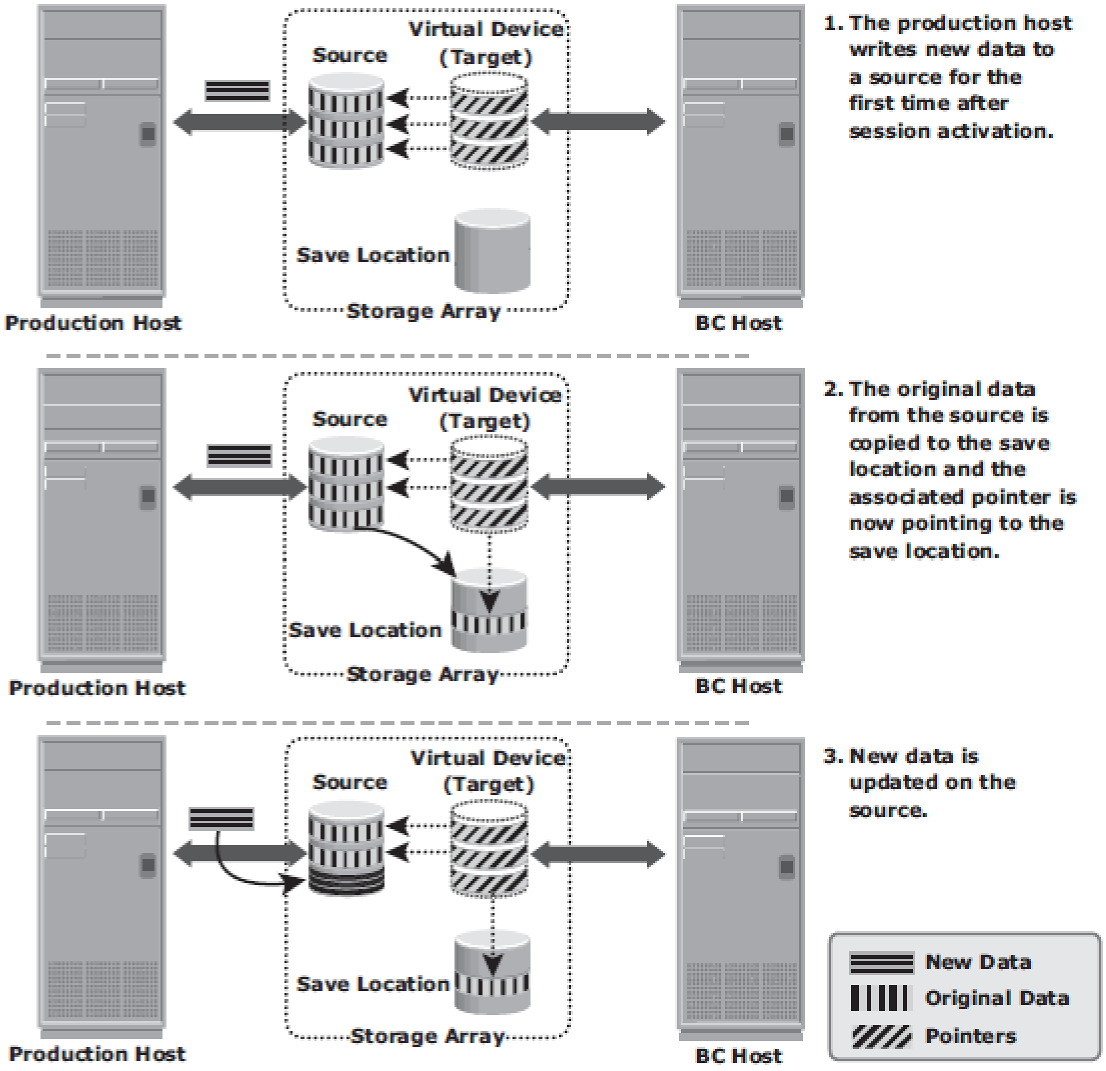

Virtual Device
Source
(Target)
1. The production host writes new data to a source for the first time after session activation.
Save Location
Storage Array
Production Host
BC Host
2. The original data from the source is copied to the save location and the associated pointer is now pointing to the save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

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

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

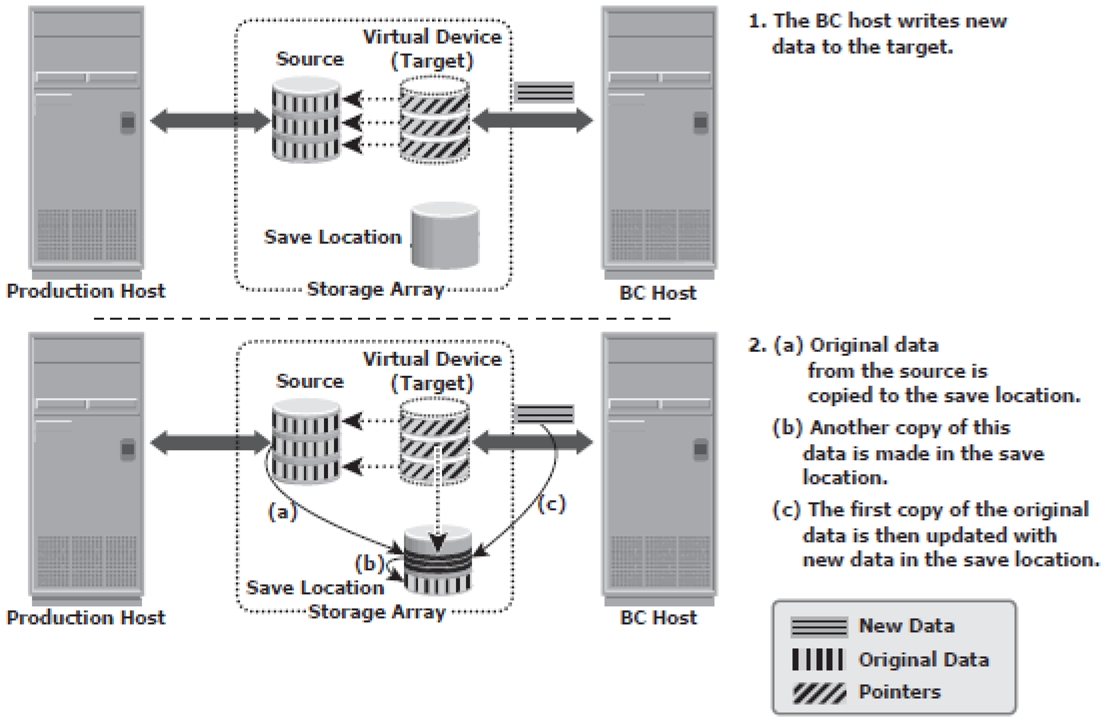

Production Host
Source
Virtual Device (Target)
Save Location
Storage Array
BC Host
1. The BC host writes new data to the target.
Production Host
Source
Virtual Device (Target)
(a)
(b)
(c)
Save Location
Storage Array
BC Host
2. (a) Original data from the source is copied to the save location.
(b) Another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

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

When reads are issued to the target, unchanged data blocks since the session activation are read from the source, whereas data blocks that have changed are read from the save location. Data on the target is a combined view of unchanged data on the source and data on the save location. Unavailability of the source device invalidates the data on the target. The target contains only pointers to the data, and therefore, the physical capacity required for the target is a fraction of the source device. The capacity required for the save location depends on the amount of the expected data change.

Network-Based Local Replication:

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

Continuous Data Protection:

- In CDP, data changes are continuously captured and stored in a separate location from the primary storage. Moreover, RPOs are random and do not need to be defined in advance.
- With CDP, recovery from data corruption poses no problem because it allows going back to a PIT image prior to the data corruption incident.
- CDP uses a journal volume to store all data changes on the primary storage.
- The journal volume contains all the data that has changed from the time the replication session started.
- The amount of space that is configured for the journal determines how far back the recovery points can go.
- CDP is typically implemented using CDP appliance and write splitters.
- CDP implementation may also be host-based, in which CDP software is installed on a separate host machine.
- CDP appliance is an intelligent hardware platform that runs the CDP software and manages local and remote data replications.
- Write splitters intercept writes to the production volume from the host and split each write into two copies.
- Write splitting can be performed at the host, fabric, or storage array.

CDP Local Replication Operation:

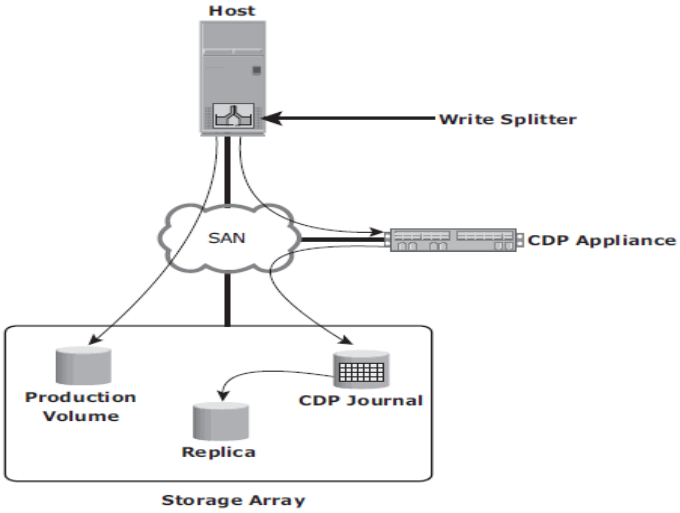


Figure 11-14: Continuous data protection — local replication

- In this method, before the start of replication, the replica is synchronized with the source and then the replication process starts.
- After the replication starts, all the writes to the source are split into two copies. One of the copies is sent to the CDP appliance and the other to the production volume.
- When the CDP appliance receives a copy of a write, it is written to the journal volume along with its timestamp.
- As a next step, data from the journal volume is sent to the replica at predefined intervals.
- While recovering data to the source, the CDP appliance restores the data from the replica and applies journal entries up to the point in time chosen for recovery.

Local Replication in a Virtualized Environment:

- In a virtualized environment, along with replicating storage volumes, virtual machine (VM) replication is also required.
- Local replication of VMs is generally performed by the hypervisor at the compute level.
- However, it can also be performed at the storage level using array-based local replication, similar to the physical environment.
- In the array-based method, the LUN on which the VMs reside is replicated to another LUN in the same array.
- For hypervisor-based local replication, two options are available: VM Snapshot and VM Clone.
- **VM Snapshot:**
    - VM Snapshot captures the state and data of a running virtual machine at a specific point in time.
    - The VM state includes VM files, such as BIOS, network configuration, and its power state (powered-on, powered-off, or suspended).
    - The VM data includes all the files that make up the VM, including virtual disks and memory.
    - A VM Snapshot uses a separate delta file to record all the changes to the virtual disk since the snapshot session is activated.
    - Snapshots are useful when a VM needs to be reverted to the previous state in the event of logical corruptions.
    - Reverting a VM to a previous state causes all settings configured in the guest OS to be reverted to that PIT when that snapshot was created.
    - The VM Snapshot technology does not support data replication if a virtual machine accesses the data by using raw disks. Also, using the hypervisor to

perform snapshots increases the load on the compute and impacts the compute performance.

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

Remote Replication Technologies

Host-Based Remote Replication

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

LVM-Based Remote Replication:

- LVM-based remote replication is performed and managed at the volume group level.
- Writes to the source volumes are transmitted to the remote host by the LVM.
- The LVM on the remote host receives the writes and commits them to the remote volume group.
- Prior to the start of replication, identical volume groups, logical volumes, and file systems are created at the source and target sites.
- Initial synchronization of data between the source and replica is performed.
- One method to perform initial synchronization is to backup the source data and restore the data to the remote replica. Alternatively, it can be performed by replicating over the IP network.
- Until the completion of the initial synchronization, production work on the source volumes is typically halted.
- After the initial synchronization, production work can be started on the source volumes and replication of data can be performed over an existing standard IP network

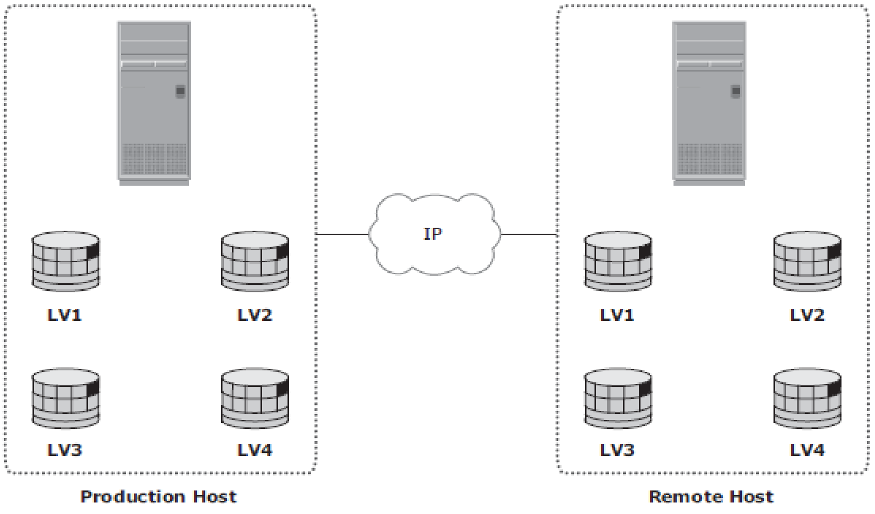


Figure 12-5: LVM-based remote replication

LVM-based remote replication is independent of the storage arrays and therefore supports replication between heterogeneous storage arrays. Most operating systems are shipped with LVMs, so additional licenses and specialized hardware are not typically required.

The replication process adds overhead on the host CPUs. CPU resources on the source host are shared between replication tasks and applications. This might cause performance degradation to the applications running on the host. Because the remote host is also involved in the replication process, it must be continuously up and available.

Host-Based Log Shipping:

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

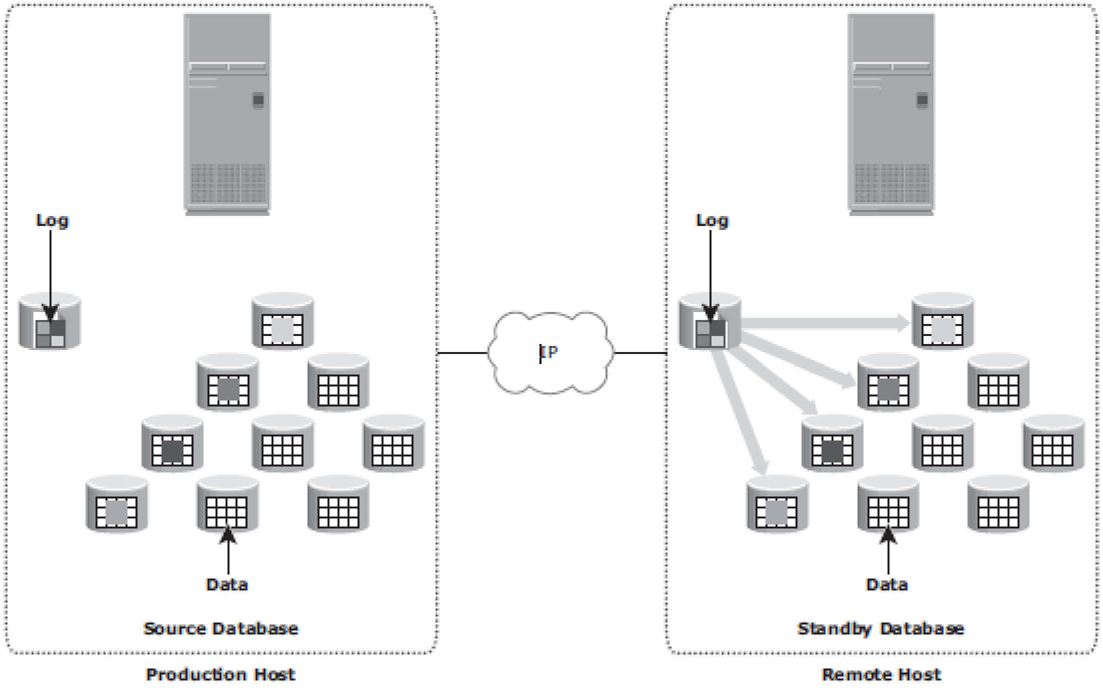


Figure 12-6: Host-based log shipping

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

- The remote database is started in a standby mode. Typically, in standby mode, the database is not available for transactions.
- All DBMSs switch log files at preconfigured time intervals or when a log file is full.
- The current log file is closed at the time of log switching, and a new log file is opened.
- When a log switch occurs, the closed log file is transmitted by the source host to the remote host.
- The remote host receives the log and updates the standby database. This process ensures that the standby database is consistent up to the last committed log.
- RPO at the remote site is finite and depends on the size of the log and the frequency of log switching. Available network bandwidth, latency, rate of updates to the source database, and the frequency of log switching should be considered when determining the optimal size of the log file.
- Similar to LVM-based remote replication, the existing standard IP network can be used for replicating log files. Host-based log shipping requires low network bandwidth because it transmits only the log files at regular intervals.

Storage Array-Based Remote Replication:

In storage array-based remote replication, the array-operating environment and resources perform and manage data replication. This relieves the burden on the host CPUs, which can be better used for applications running on the host.

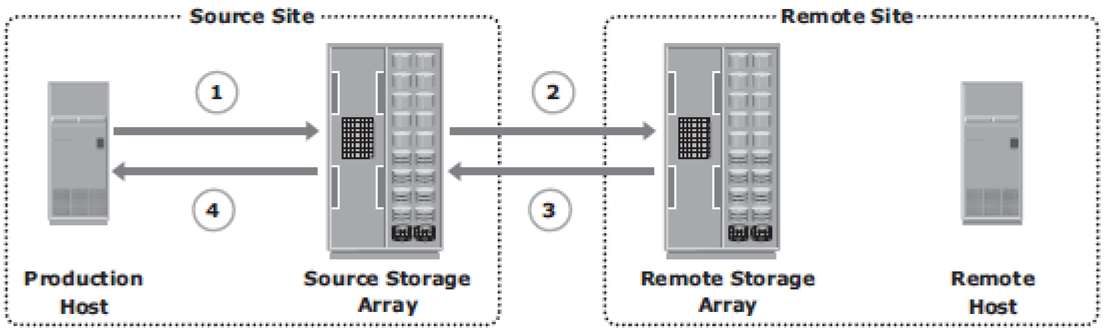
A source and its replica device reside on different storage arrays. Data can be transmitted from the source storage array to the target storage array over a shared or a dedicated network.

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

Synchronous Replication Mode:

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

- In the case of synchronous remote replication, to optimize the replication process and to minimize the impact on application response time, the write is placed on cache of the two arrays.
- The intelligent storage arrays destage these writes to the appropriate disks later.
- If the network links fail, replication is suspended; however, production work can continue uninterrupted on the source storage array.
- The array operating environment keeps track of the writes that are not transmitted to the remote storage array.
- When the network links are restored, the accumulated data is transmitted to the remote storage array.
- During the time of network link outage, if there is a failure at the source site, some data will be lost, and the RPO at the target will not be zero.

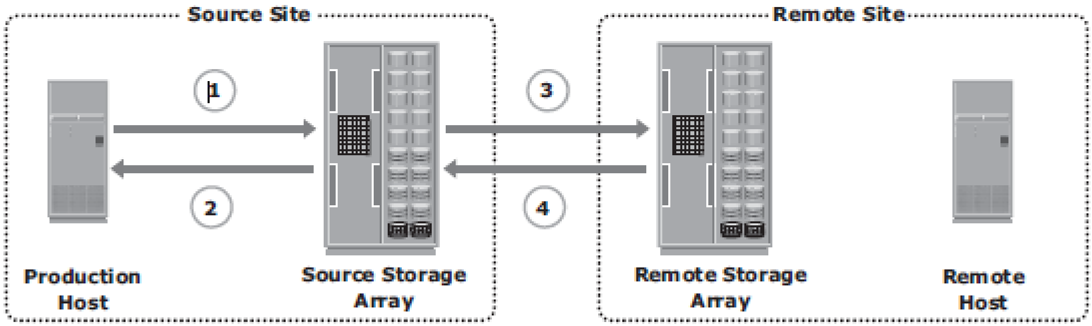


1. Write from the production host is received by the source storage array.
2. Write is then transmitted to the remote storage array.
3. Acknowledgment is sent to the source storage array by the remote storage array.
4. Source storage array signals write-completion to the production host.

Figure 12-7: Array-based synchronous remote replication

Asynchronous Replication Mode:

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



1 The production host writes to the source storage array.

2 The source array immediately acknowledges the production host.

3 These writes are then transmitted to the target array.

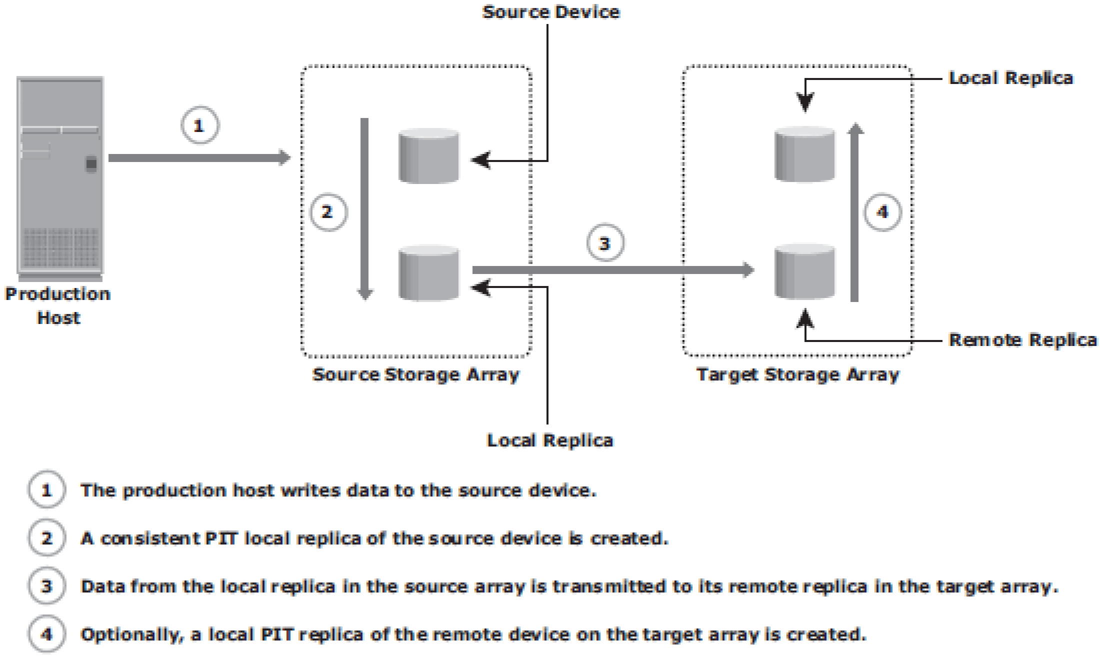
4 After the writes are received by the target array, it sends an acknowledgment to the source array.

Figure 12-8: Array-based asynchronous remote replication

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

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

Disk-Buffered Replication Mode:



1. The production host writes data to the source device.
2. A consistent PIT local replica of the source device is created.
3. Data from the local replica in the source array is transmitted to its remote replica in the target array.
4. Optionally, a local PIT replica of the remote device on the target array is created.

Figure 12-9: Disk-buffered remote replication

- Disk-buffered replication is a combination of local and remote replication technologies.
- A consistent PIT local replica of the source device is first created. This is then replicated to a remote replica on the target array.
- Figure 12-9 shows the sequence of operations in a disk-buffered remote replication.
- At the beginning of the cycle, the network links between the two arrays are suspended, and there is no transmission of data. While production application runs on the source device, a consistent PIT local replica of the source device is created.
- The network links are enabled, and data on the local replica in the source array transmits to its remote replica in the target array. After synchronization of this pair, the network link is suspended, and the next local replica of the source is created.

- Optionally, a local PIT replica of the remote device on the target array can be created. The frequency of this cycle of operations depends on the available link bandwidth and the data change rate on the source device.
- Because disk-buffered technology uses local replication, changes made to the source and its replica are possible to track. Therefore, all the resynchronization operations between the source and target can be done incrementally.
- When compared to synchronous and asynchronous replications, disk-buffered remote replication requires less bandwidth.

Network-Based Remote Replication:

CDP Remote Replication:

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

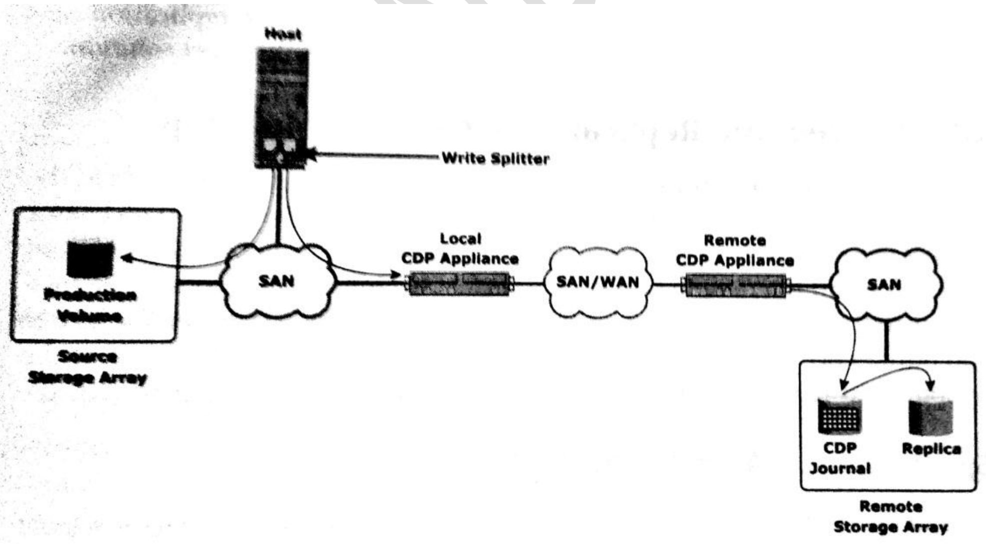


Figure 12-10: CDP remote replication

Figure 12-10 describes CDP remote replication. In this method, the replica is synchronized with the source, and then the replication process starts.

After the replication starts, all the writes from the host to the source are split into two copies. One of the copies is sent to the local CDP appliance at the source site, and the other copy is sent to the production volume.

After receiving the write, the appliance at the source site sends it to the appliance at the remote site. Then, the write is applied to the journal volume at the remote site. For an asynchronous operation, writes at the source CDP appliance are accumulated, and redundant blocks are eliminated. Then, the writes are sequenced and stored with their corresponding timestamp.

The data is then compressed, and a checksum is generated. It is then scheduled for delivery across the IP or FC network to the remote CDP appliance. After the data is received, the remote appliance verifies the checksum to ensure the integrity of the data. The data is then uncompressed and written to the remote journal volume.

As a next step, data from the journal volume is sent to the replica at predefined intervals. In the asynchronous mode, the local CDP appliance instantly acknowledges write as soon as it is received.

In the synchronous replication mode, the host application waits for an acknowledgment from the CDP appliance at the remote site before initiating the next write. The synchronous replication mode impacts the application's performance under heavy write loads.

For remote replication over extended distances, optical network technologies such as dense wavelength division multiplexing (DWDM), course wavelength division multiplexing (WDM), and synchronous optical network (SONET) are deployed.

Three-Site Replication:

Three-site replication mitigates the risks identified in two-site replication. In a three-site replication, data from the source site is replicated to two remote sites.

Replication can be synchronous to one of the two sites, providing near zero-RPO solution, and it can be asynchronous or disk buffered to the other remote site, providing a finite RPO. Three-site remote replication can be implemented as a **cascade/multihop or a triangle/multitarget solution.**

Three-Site Replication — Cascade/Multihop:

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

Replication between the source and the remotesites can be performed in two ways: synchronous + asynchronous or synchronous + disk buffered.

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

Synchronous + Asynchronous:

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

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

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

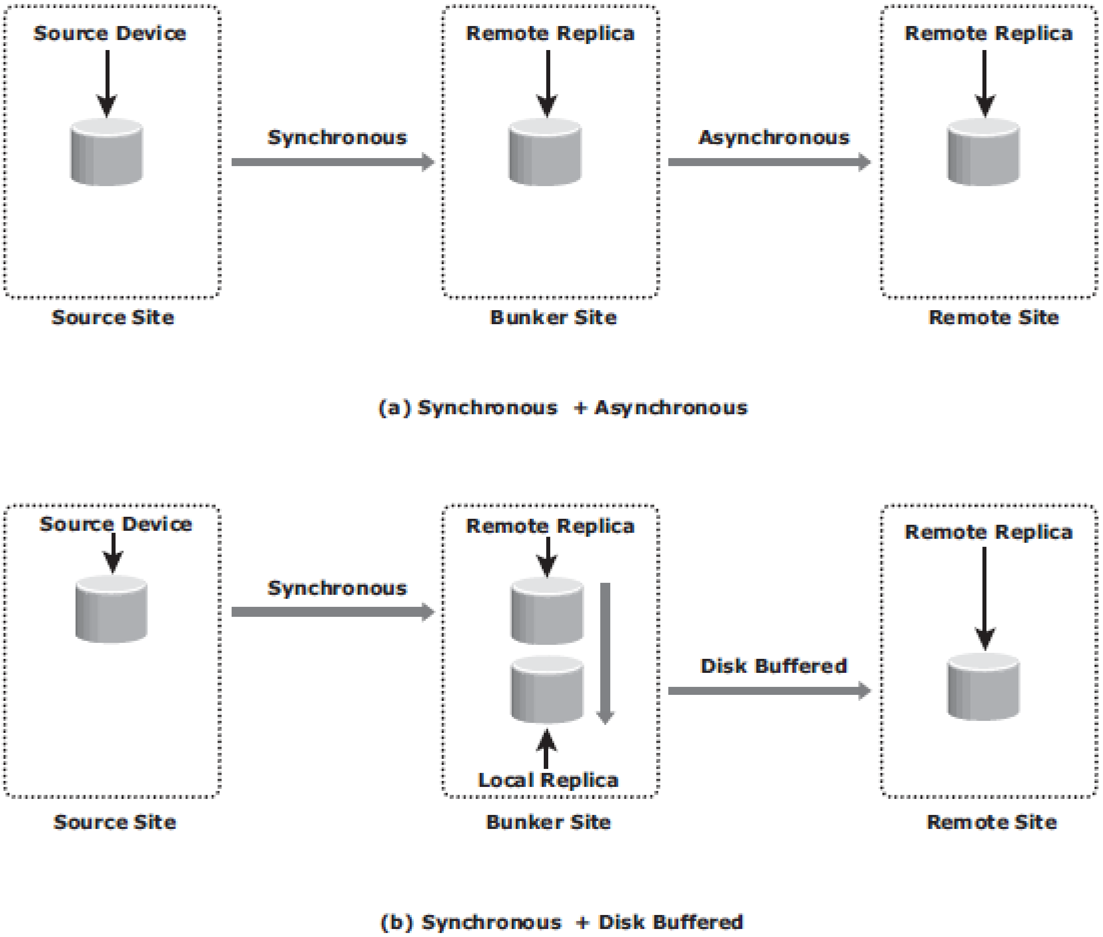


Figure 12-11: Three-site remote replication cascade/multihop

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

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

If a disaster occurs at the remote site, or if the network links between the bunker and the remote site fail, the source site continues to work as normal with disaster recovery protection provided at the bunker site.

Synchronous + Disk Buffered:

This method employs a combination of local and remote replication technologies. Synchronous replication occurs between the source and the bunker: a consistent PIT local replica is created at the bunker. Data is transmitted from the local replica at the bunker to the remote replica at the remote site.

Optionally, a local replica can be created at the remote site after data is received from the bunker. Figure 12-11 (b) illustrates the synchronous + disk buffered method.

In this method, a minimum of four storage devices are required (including the source) to replicate one storage device. The other three devices are the synchronous remote replica at the bunker, a consistent PIT local replica at the bunker, and the replica at the remote site. RPO at the remote site is usually in the order of hours for this implementation. The process to create the consistent PIT copy at the bunker and incrementally updating the remote replica occurs continuously in a cycle.

Three-Site Replication — Triangle/Multitarget:

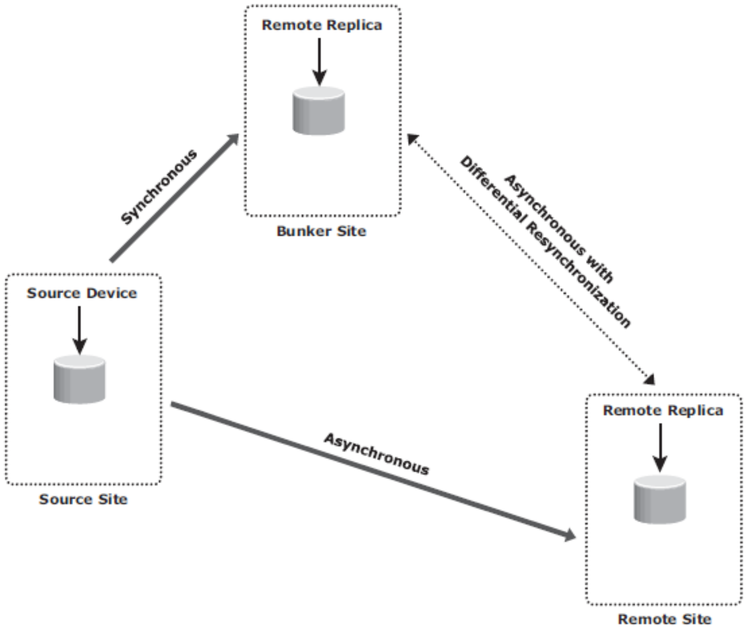


Figure 12-12: Three-site replication triangle/multitarget

In three-site triangle/multitarget replication, data at the source storage array is concurrently replicated to two different arrays at two different sites, as shown in Figure 12-12. The source-to-bunker site (target 1) replication is synchronous with a near-zero RPO. The source-to-remote site (target 2) replication is asynchronous with an RPO in the order of minutes.

The distance between the source and the remote sites could be thousands of miles. This implementation does not depend on the bunker site for updating data on the remote site because data is asynchronously copied to the remote site directly from the source. The triangle/multitarget configuration provides consistent RPO unlike cascade/ multihop solutions in which the failure of the bunker site results in the remote site falling behind and the RPO increasing.

The key benefit of three-site triangle/multitarget replication is the ability to failover to either of the two remote sites in the case of source-site failure, with disaster recovery (asynchronous) protection between the bunker and remote sites. Resynchronization between the two surviving target sites is incremental. Disaster recovery protection is always available if any one-site failure occurs.

During normal operations, all three sites are available and the production workload is at the source site. At any given instant, the data at the bunker and the source is identical. The data at the remote site is behind the data at the source and the bunker. The replication network links between the bunker and remote sites will be in place but not in use. Thus, during normal operations, there is no data movement between the bunker and remote arrays. The difference in the data between the bunker and remote sites is tracked so that if a source site disaster occurs, operations can be resumed at the bunker or the remote sites with incremental resynchronization between these two sites.

A regional disaster in three-site triangle/multitarget replication is similar to a source site failure in two-site asynchronous replication. If failure occurs, operations failover to the remote site with an RPO within minutes. There is no remote protection until the regional disaster is resolved. Local replication technologies could be used at the remote site during this time.

A failure of the bunker or the remote site is not actually considered a disaster because the operation can continue uninterrupted at the source site while remote disaster recovery protection is still available.

A network link failure to either the source-to-bunker or the source-to-remote site does not impact production at the source site while remote disaster recovery protection is still available with the site that can be reached.

Remote Replication and Migration in a Virtualized Environment:

In a virtualized environment, all VM data and VM configuration files residing on the storage array at the primary site are replicated to the storage array at the remote site. This process remains transparent to the VMs.

The LUNs are replicated between the two sites using the storage array replication technology. This replication process can be either synchronous (limited distance, near zero RPO) or asynchronous (extended distance, nonzero RPO).

Virtual machine migration is another technique used to ensure business continuity in case of hypervisor failure or scheduled maintenance. **VM migration is the process to move VMs from one hypervisor to another without powering off the virtual machines.**

VM migration also helps in load balancing when multiple virtual machines running on the same hypervisor contend for resources. Two commonly used techniques for VM migration are **hypervisor-to-hypervisor and array-to-array migration.**

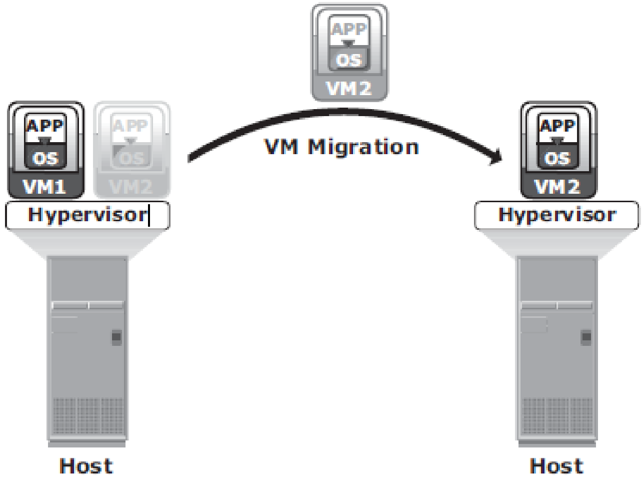


Figure 12-14: Hypervisor-to-hypervisor VM migration

In hypervisor-to-hypervisor VM migration, the entire active state of a VM is moved from one hypervisor to another. Figure 12-14 shows hypervisor-to hypervisorVM migration. This method involves copying the contents of virtual machine memory from the source hypervisor to the target and then transferring the control of the VM's disk files to the target hypervisor. Because the virtual disks of the VMs are not migrated, this technique requires both source and target hypervisor access to the same storage.

In array-to-array VM migration, virtual disks are moved from the source array to the remote array. This approach enables the administrator to move VMs across dissimilar storage arrays. Figure 12-15 shows array-to-array VM migration.

Array-to-array migration starts by copying the metadata about the VM from the source array to the target. The metadata essentially consists of configuration, swap, and log files. After the metadata is copied, the VM disk file is replicated to the new location.

During replication, there might be a chance that the source is updated; therefore, it is necessary to track the changes on the source to maintain data integrity. After the replication is complete, the blocks that have changed since the replication started are replicated to the new location.

Array-to-array VM migration improves performance and balances the storage capacity by redistributing virtual disks to different storage devices.

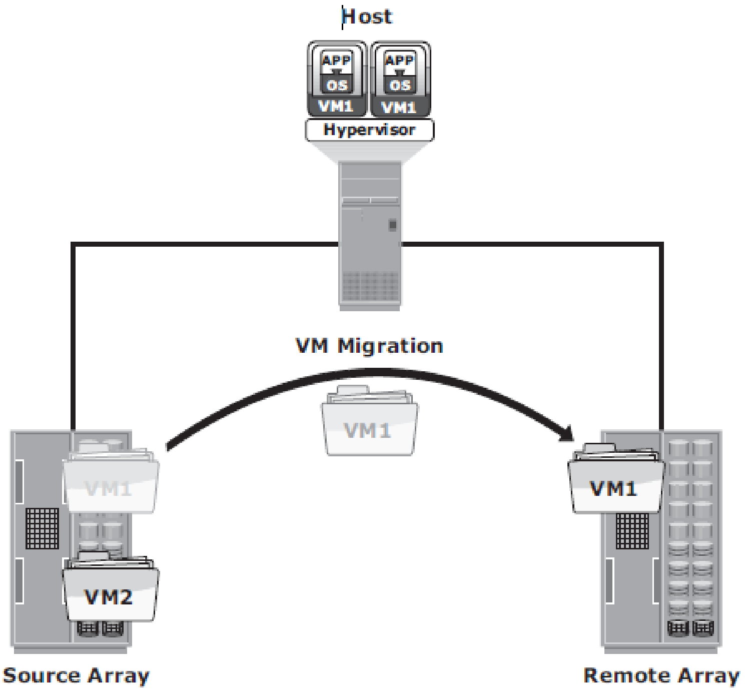


Figure 12-15: Array-to-array VM migration