FUTURE VISION BIE

One Stop for All Study Materials

& Lab Programs

Future Vision

By K B Hemanth Raj

Scan the QR Code to Visit the Web Page

Or

Visit : https://hemanthrajhemu.github.io

Gain Access to All Study Materials according to VTU,
CSE – Computer Science Engineering,
ISE – Information Science Engineering,
ECE - Electronics and Communication Engineering
& MORE...

Join Telegram to get Instant Updates: https://bit.ly/VTU_TELEGRAM

Contact: MAIL: futurevisionbie@gmail.com

INSTAGRAM: www.instagram.com/hemanthraj_hemu/

INSTAGRAM: www.instagram.com/futurevisionbie/

WHATSAPP SHARE: https://bit.ly/FVBIESHARE

BMS
Institute of Technology and Management

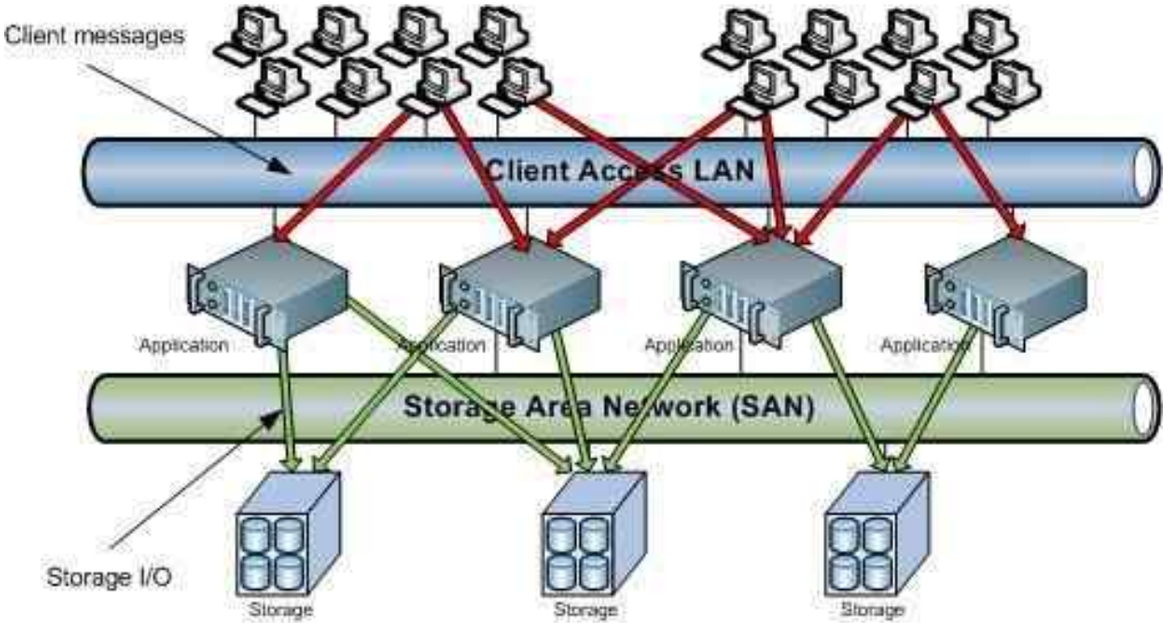
Avalahalli, Doddaballapur Main Road, Bengaluru – 560064

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Storage Area Networks

(17CS754)

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



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

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

CREDITS – 03

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

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

Course outcomes: The students should be able to:

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

Question paper pattern:

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

Text Books:

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

Table of Content

Module-3

Backup, Archive, and Replication

Information Availability

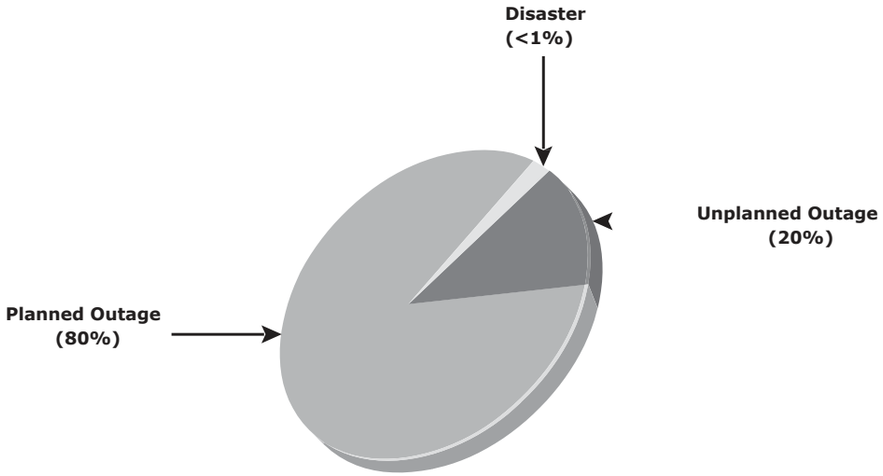
Information availability (IA) refers to the ability of an IT infrastructure to function according to business expectations during its specified time of operation. IA ensures that people (employees, customers, suppliers, and partners) can access information whenever they need it. IA can be defined in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

- **Accessibility:** Information should be accessible at the right place, to the right user.
- **Reliability:** Informationshouldbereliable andcorrectin all aspects. It is —the samell as what was stored, and there is no alteration or corruption to the information.
- **Timeliness:** Defines the exact moment or the time window (a particu- lar 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 consid- ered 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 vir- tual components.

Another type of incident that may cause data unavailability is natural or man- madedisasters, such as flood, fire, earthquake, and contamination. As illustrated in - 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.



- 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 withcustomers, 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)

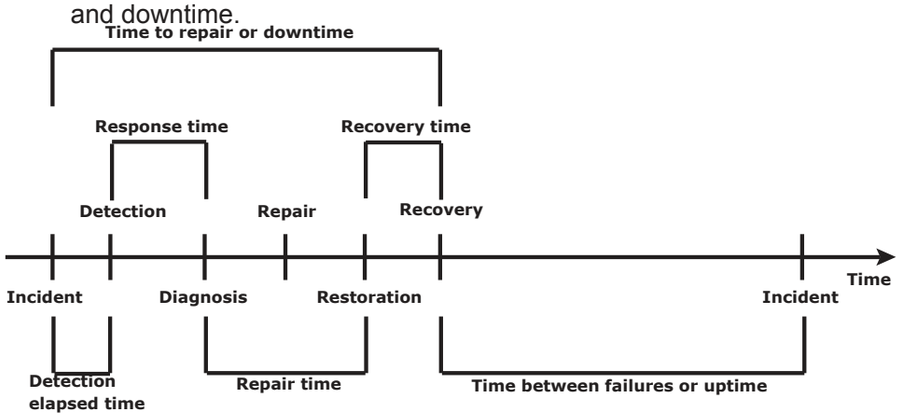
Average revenue loss per hour = (total revenue of an organization per week)/(average number of hours per week that an organization is open for business)

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 componentsmight 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 involvesrestoringacomponenttoaconditionthat 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 mea- sured 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. - 9-2 illustrates the various information availability metrics that represent system uptime and downtime.



- 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. Thiscalculationleadstothe number of —9sll representationforavailabilitymetrics. 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 availablell is available for 99.999 percent of the scheduled time in a year (24 × 365).

Table 9-1: Availability Percentage and Allowable Downtime

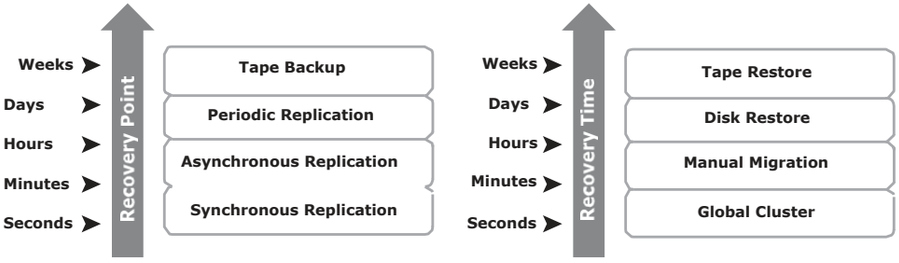
UPTIME (%)	DOWNTIME (%)	DOWNTIME PER YEAR	DOWNTIME PER WEEK
98	2	7.3 days	3 hr, 22 minutes
99	1	3.65 days	1 hr, 41 minutes
99.8	0.2	17 hr, 31 minutes	20 minutes, 10 secs
99.9	0.1	8 hr, 45 minutes	10 minutes, 5 secs
99.99	0.01	52.5 minutes	1 minute
99.999	0.001	5.25 minutes	6 secs
99.9999	0.0001	31.5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

BC Terminology

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

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

- **Disasterrestart:** Thisistheprocessofrestartingbusinessoperationswith 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. - 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. Thecorresponding recoverystrategyistorecoverthedatabase tothe point of the last log shipment.
  - **RPO in the order of minutes:** Mirroring data asynchronously to a remote site
  - **Near zero RPO:** Mirroring data synchronously to a remote site



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

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

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

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

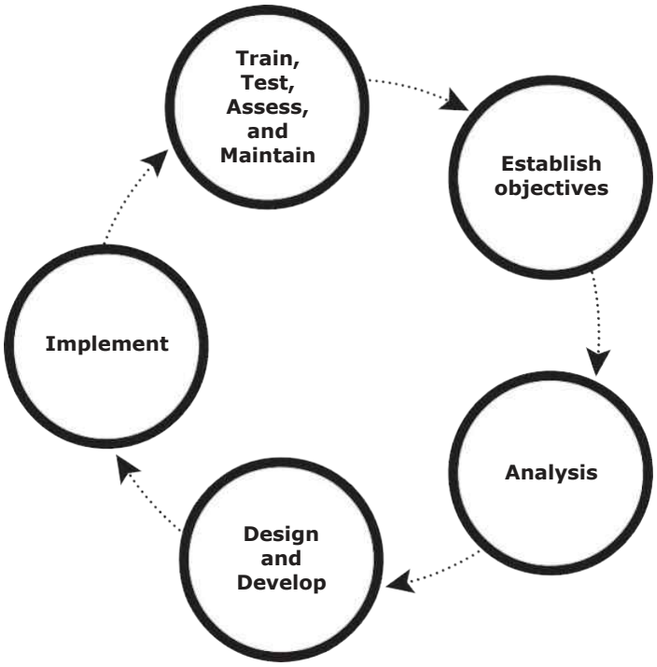
- **RTO of 72 hours:** Restore from tapes available at a cold site.
- **RTOof12hours:**Restore fromtapes available atahotsite.
- **RTO of few hours:** Use of data vault at a hot site
- **RTO of a few seconds:** Cluster production servers with bidirectional mirroring, enablingtheapplicationstorunatbothsitessimultaneously.

- **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 availabil- ity 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, mul- tiple servers may access data, and typically one server is kept as standby. Server clustering provides load balancing by distributing the application load evenly among multiple servers within the cluster.

BC Planning Life Cycle

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

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



- 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.
   - CreateBCpolicies.
2. Analysis:
   - Collect information on data profiles, business processes, infrastructure support,dependencies, andfrequencyofusingbusinessinfrastructure.
   - Conduct a Business Impact Analysis (BIA).
   - Identify critical business processes and assign recovery priorities.
   - Perform risk analysis for critical functions and create mitigation strategies.

- Performcostbenefitanalysisforavailable solutionsbased onthemitigation 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 asemergency response, damage assessment, andinfrastructure 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 datacenter.
   - Implementredundancyforeveryresourceinadatacentertoavoid single points of failure.

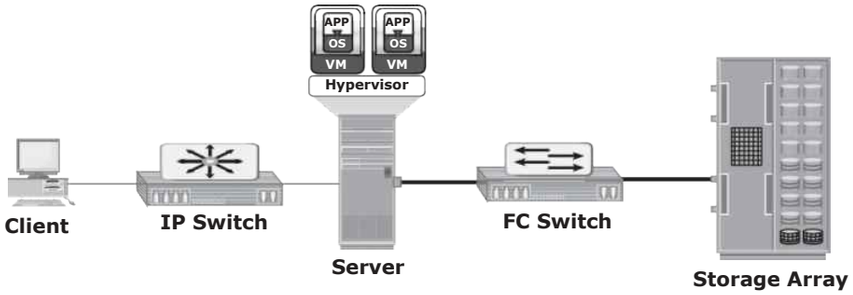
5. Train, test, assess, and maintain:
   - Traintheemployeeswhoareresponsibleforbackupandreplicationof business-critical data on a regular basis or whenever there is amodification in the BC plan.
   - Train employees on emergency response procedures when disasters are declared.
   - Traintherecoveryteamonrecoveryproceduresbasedoncontingency scenarios.
   - Perform damage-assessment processes and review recovery plans.
   - Testthe BCplanregularlytoevaluateitsperformanceandidentify its limitations.
   - Assess the performance reports and identify limitations.
   - Update the BC plans and recovery/restart procedures to reflect regular changes within the datacenter.

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. - 9-5 depicts a system setup in which an application, running on a VM, provides an interface to the client and performs I/Ooperations. The client is connected to the server through an IP network, and the server is connected to the storage array through an FC connection.



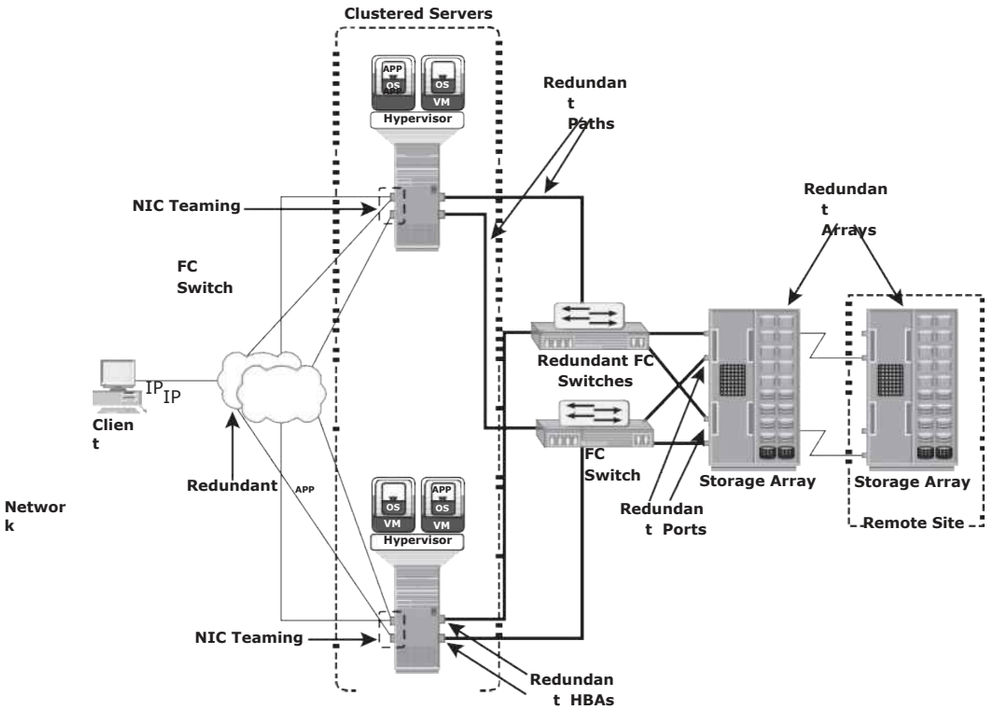
- 9-5: Single point of failure

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 - 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 - 9-6 represents all enhancements in the infrastructure to mitigate single points of failure:

- ConfigurationofredundantHBAsataservertomitigatesingleHBAfailure
- 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.



- 9-6: Resolving single points of failure

Multipathing Software

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

In practice, merely configuring multiple paths does not serve the purpose. Even with multiple paths, if one path fails, I/Odoes not reroute unless the system recog- nizes that it has an alternative path. Multipathing software provides the function- ality to recognize and utilize alternative I/Opaths 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 set- ting 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 infra- structure 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 attributesof the business processin terms of applications, databases, and hardware and software requirements.
- Estimate the costs of failure for each business process.
- Calculatethemaximumtolerableoutageanddefine RTOand RPOfor 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 alterna- tive 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.
- **Localreplication:** Datacanbereplicatedtoaseparatelocationwithinthe samestoragearray. Thereplicaisusedindependentlyforotherbusiness operations. Replicascanalsobeusedforrestoringoperations if datacor- ruption occurs.
- **Remote replication:** Data in a storage array can be replicated to another storage array located at a remote site. If the storage array is lost due to a disaster, business operations can be started from the remote storage array.

Concept in Practice: EMC PowerPath

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

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

PowerPath Features

PowerPath provides the following features:

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

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

- **Cluster support:** The deployment of PowerPathin aserver cluster eliminates invoking cluster failover due to a path failure.

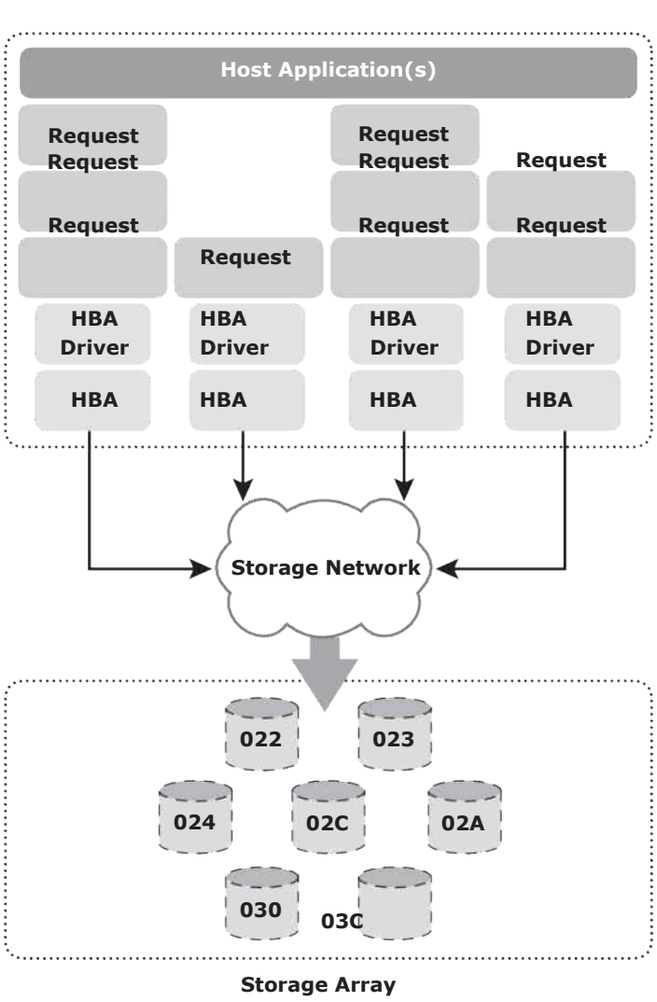
Dynamic Load Balancing

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

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

I/O Operation without PowerPath

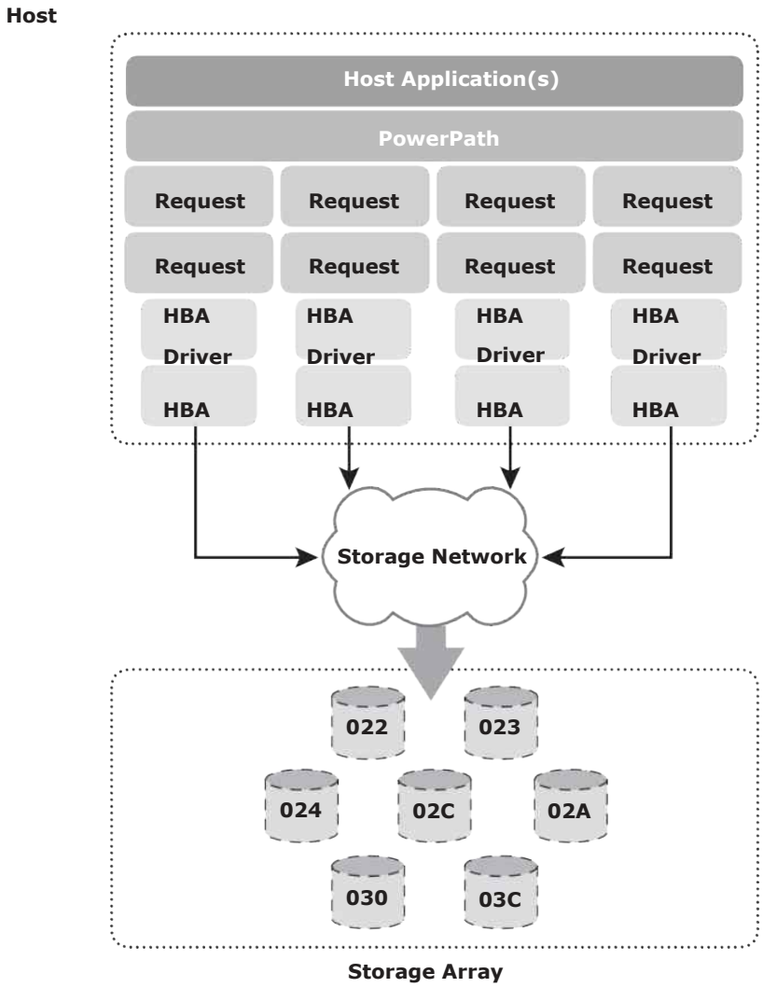
- 9-7 illustrates I/O operations in a storage system in the absence of PowerPath. The applications running on a host have four paths to the storage array. This example illustrates how I/O throughput is unbalanced without PowerPath. Two paths get high I/O traffic and are highly loaded, whereas the other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



- 9-7: I/O without PowerPath

I/O Operation with PowerPath

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



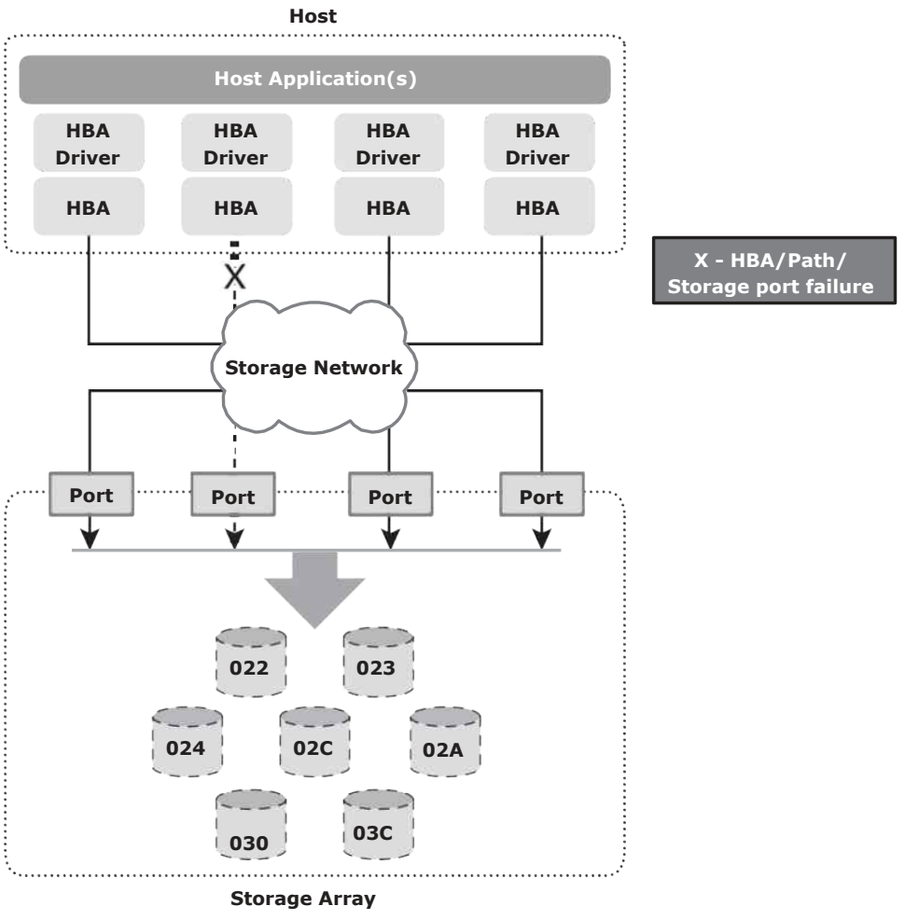
- 9-8: I/O with PowerPath

Automatic Path Failover

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

Path Failure without PowerPath

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

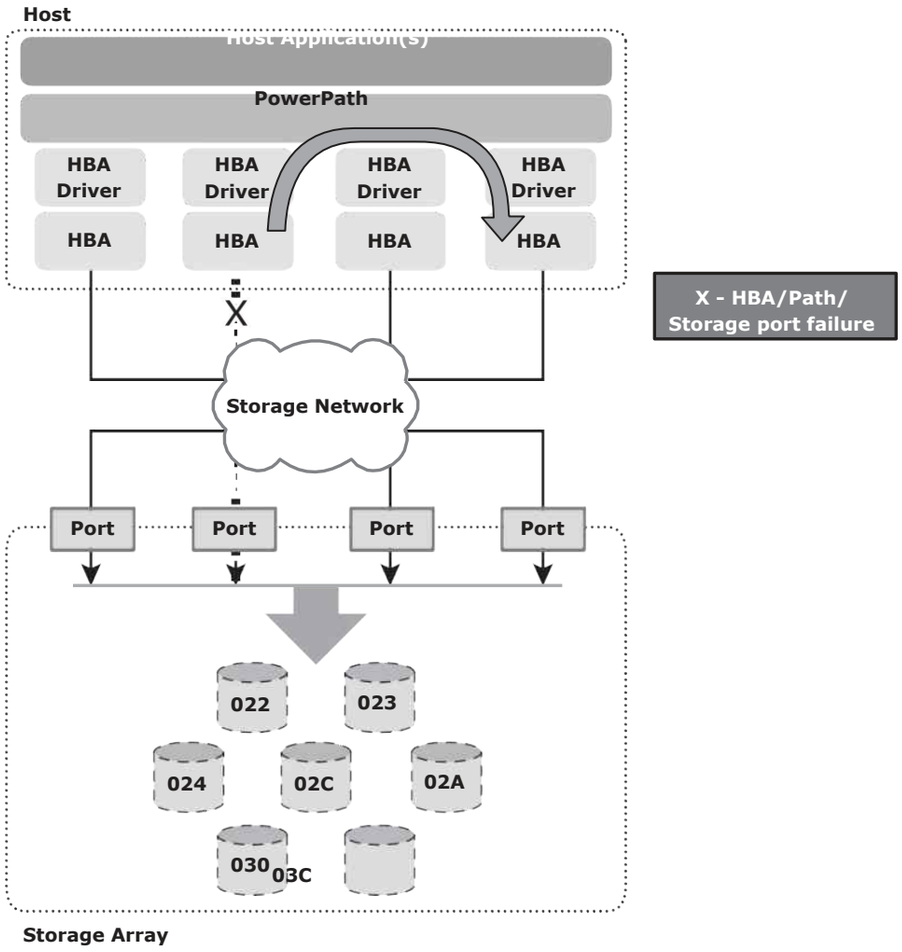


- 9-9: Path failure without PowerPath

Path Failover with PowerPath: Active-Active Array

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

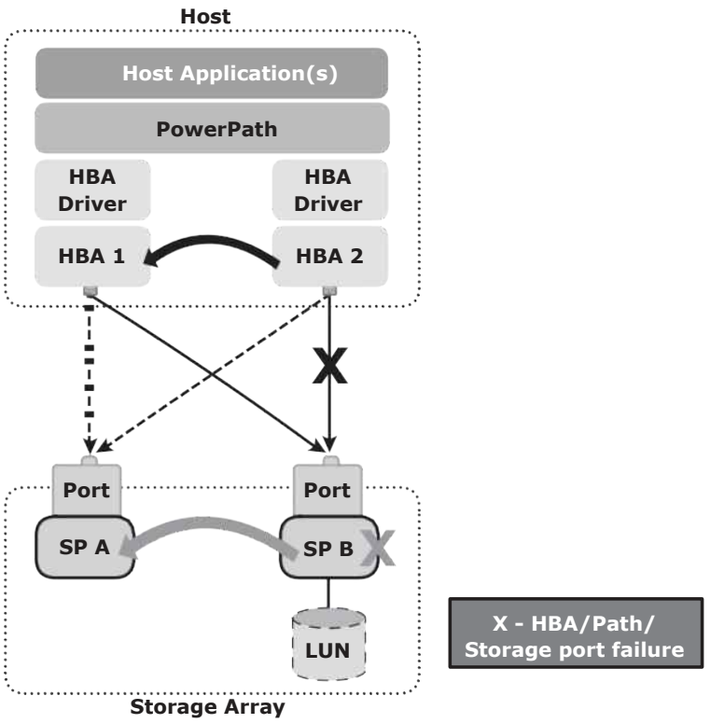
all are active and provide access to the device. If a path to the device fails, PowerPath redirects the application I/Os through an alternative active path therefore preventing any application outage.



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

Path Failover with PowerPath: Active-Passive Array

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



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

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

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

Backup Purpose

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

Disaster Recovery

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

Operational Recovery

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

Archival

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

Backup Considerations

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

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

Organizations must also consider the granularity of backups, explained later

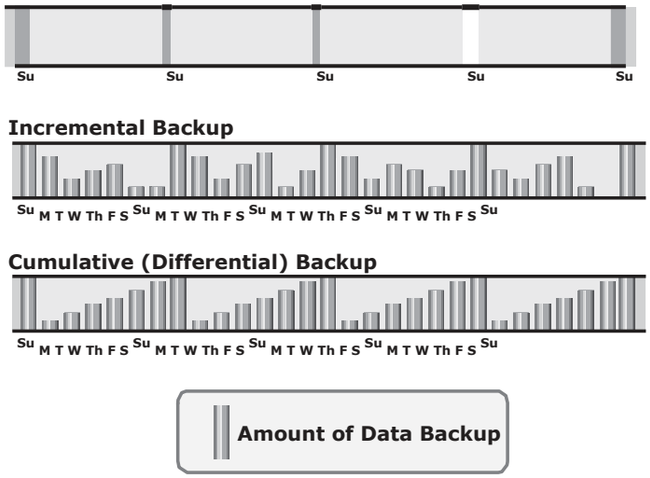
in section —10.3 Backup Granularity.‖ The development of a backup strategy must include a decision about the most appropriate time for performing a backup to minimize any disruption to production operations. The location, size, number of files, and data compression should also be considered because they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

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

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

Backup Granularity

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

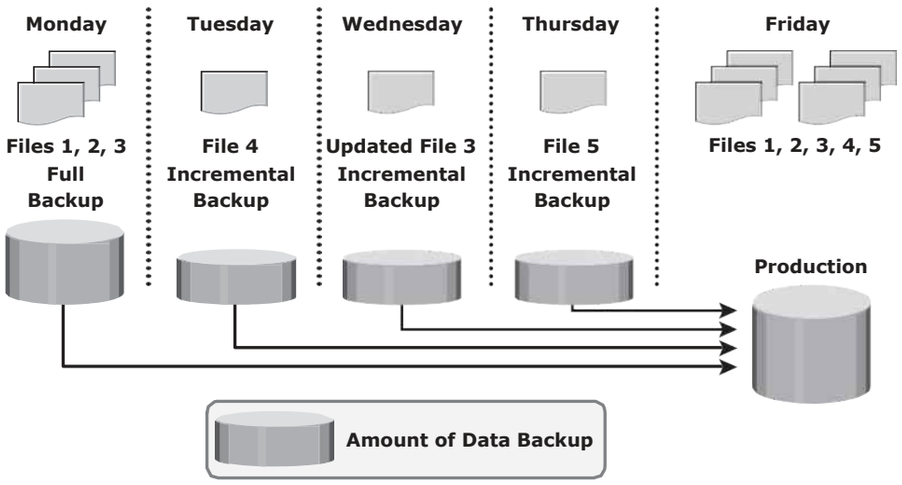


- 10-1: Backup granularity levels

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

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

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

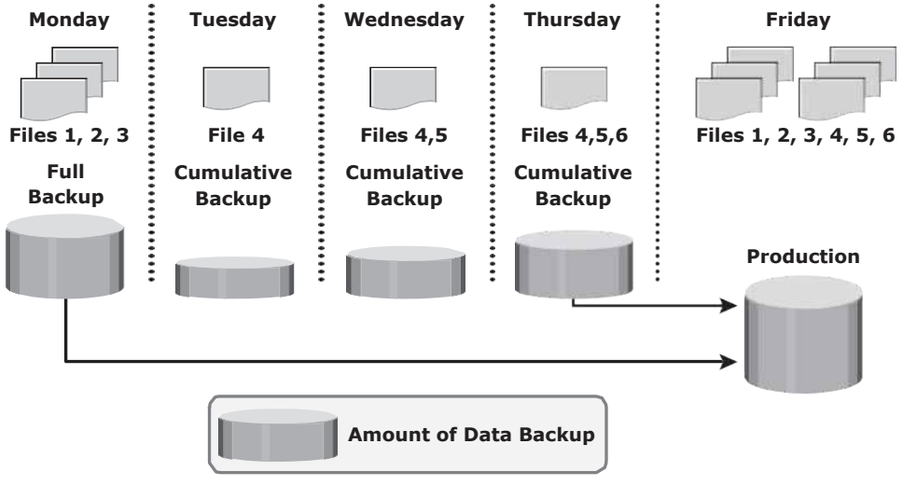


- 10-2: Restoring from an incremental backup

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

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

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



- 10-3: Restoring a cumulative backup

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

Recovery Considerations

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

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

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

Backup Methods

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

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

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

To ensure consistency, it is not enough to back up only the production data for recovery. Certain attributes and properties attached to a file, such as permis- sions, 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, whichincludespartitioning, thefile systemlayout, theoperatingsystem, theapplica- tions, and all the relevantconfigurations. BMR recovers the base system first before starting the recovery of data files. Some BMR technologies — for example server configurationbackup(SCB) — can recovera serverevenontodissimilar hardware.

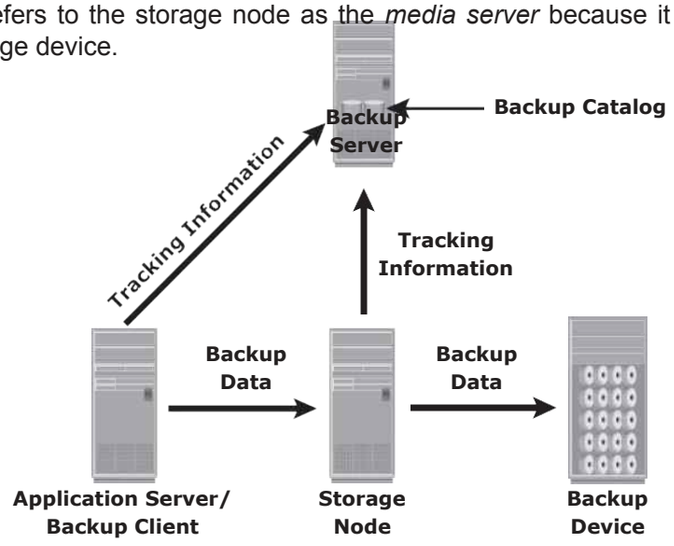
Backup Architecture

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

The storage node is responsible for writing the data to the backup device. (In a backup environment, a *storage node* is a host that controls backup devices.) The storage node also sends tracking information to the backup server. In

many cases, the storage node is integrated with the backup server, and both are hosted on the same physical platform. A backup device is attached directly

or through a network to the storage node's host platform. Some backup architecture refers to the storage node as the *media server* because it manages the storage device.



- 10-4: Backup architecture

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

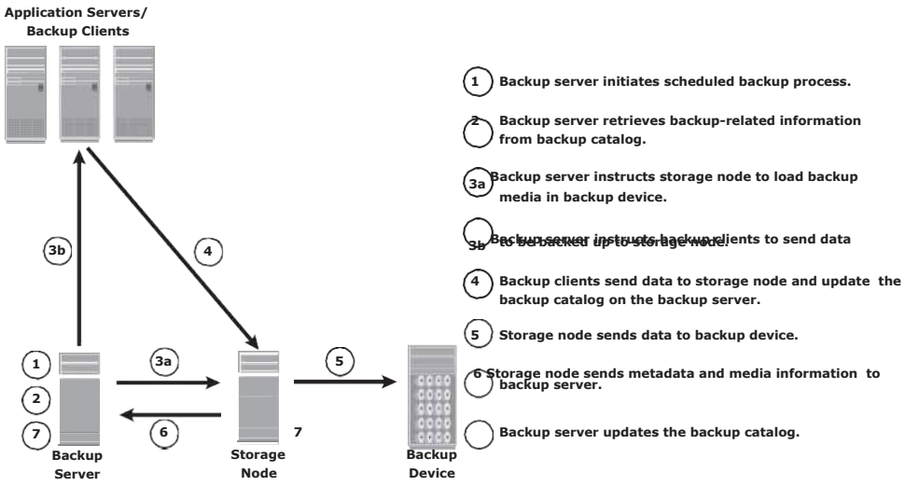
Protecting backup metadata is an important aspect of backup. If the backup catalog is lost, data recovery will be a challenge. Therefore, an updated copy of the backup cata- log should be maintained separately all the time.

Backup and Restore Operations

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

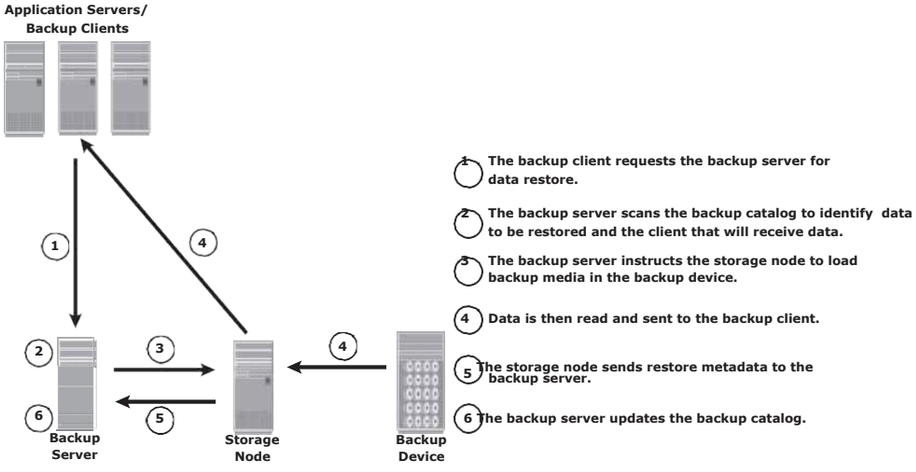
The backup server coordinates the backup process with all the components in a backup environment (see - 10-5). The backup server maintains the information about backup clients to be backed up and storage nodes to be used in a backup operation. The backup server retrieves the backup-related infor- mation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

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



- 10-5: Backup operation

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



- 10-6: Restore operation

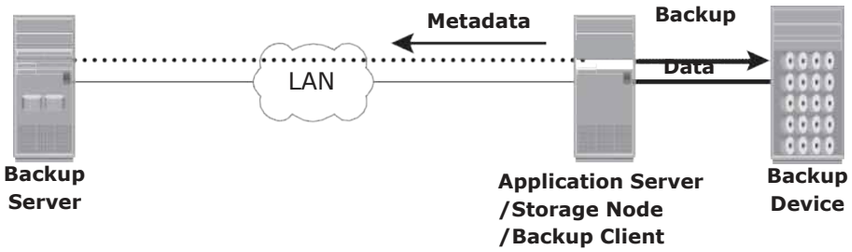
The backup server instructs the appropriate storagenode to mount the specific backup media onto the backup device. Data is then read and sent to the client that has been identified to receive the restored data.

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

Backup Topologies

Threebasictopologies are used inabackupenvironment: direct-attached backup, LAN-based backup, and SAN-based backup. A mixed topology is also used by combining LAN-based and SAN-based topologies.

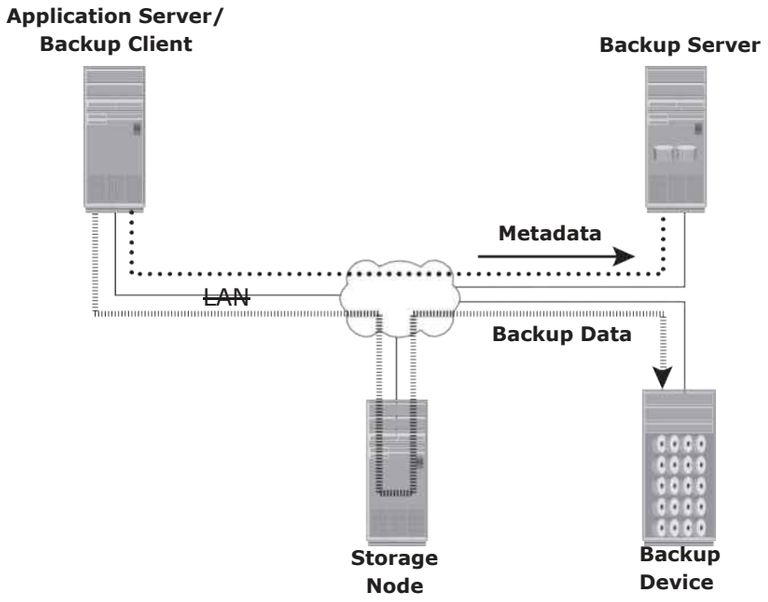
In a *direct-attached backup*, the storage node is configured on a backup client, and the backup device is attached directly to the client. Only the metadata is sent to the backup server through the LAN. This configuration frees the LAN from backup traffic. The example in - 10-7 shows that the backup device is directly attached and dedicated to the backup client. As the environ- ment 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.



- 10-7: Direct-attached backup topology

In a *LAN-based backup*, the clients, backup server, storage node, and backup device are connected to the LAN. (see - 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.



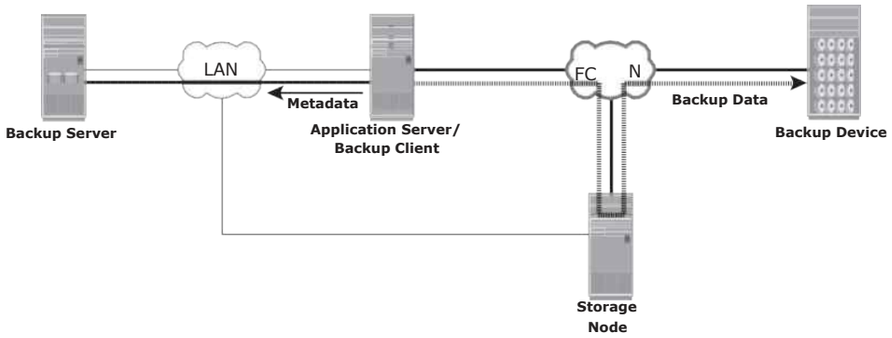
- 10-8: LAN-based backup topology

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

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

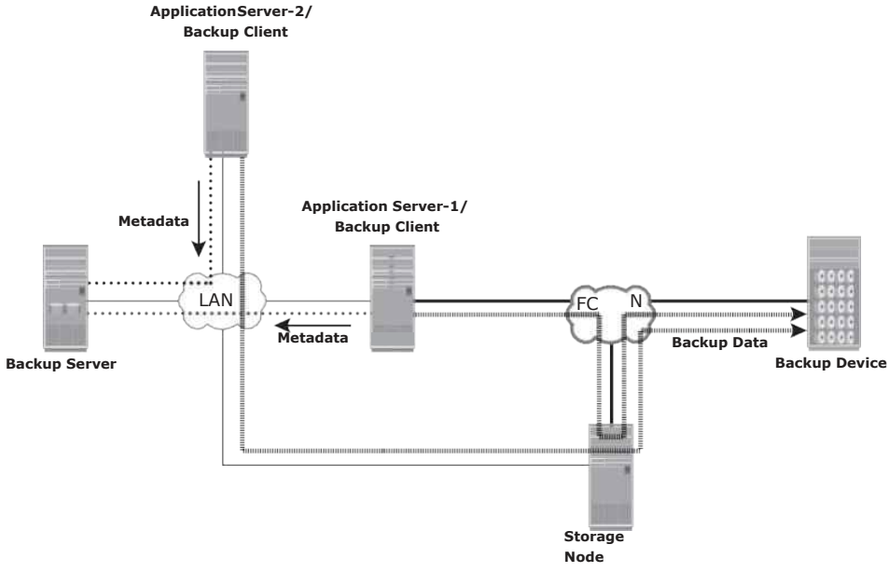
In this example, a client sends the data to be backed up to the backup device
over the SAN. Therefore, the backup data traffic is restricted to the SAN, and only the backup metadata is transported over the LAN. The volume of metadata is insignificant when compared to the production data; the LAN performance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

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



- 10-9: SAN-based backup topology

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



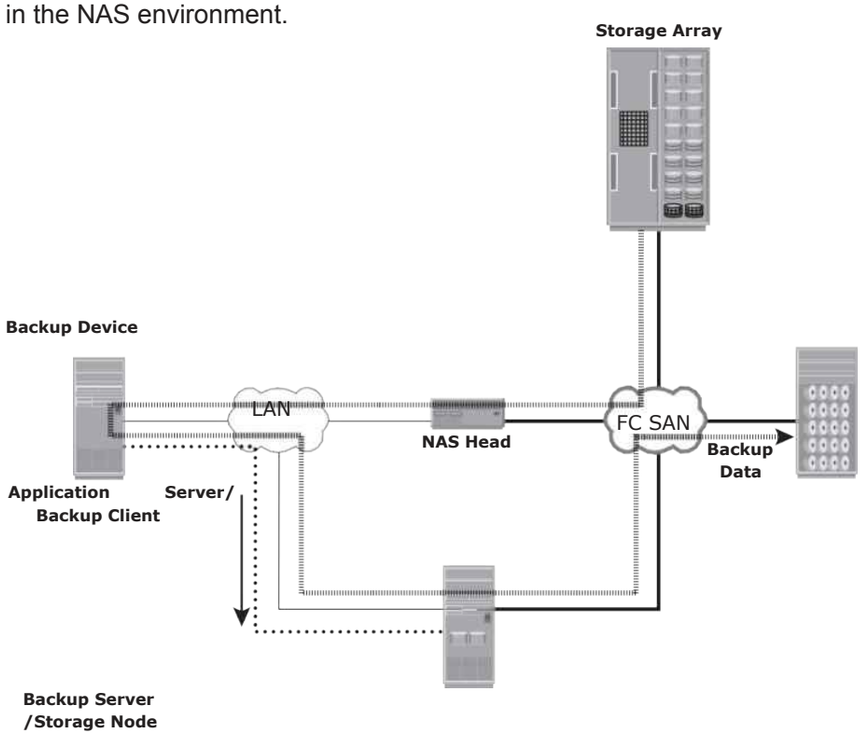
- 10-10: Mixed backup topology

Backup in NAS Environments

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

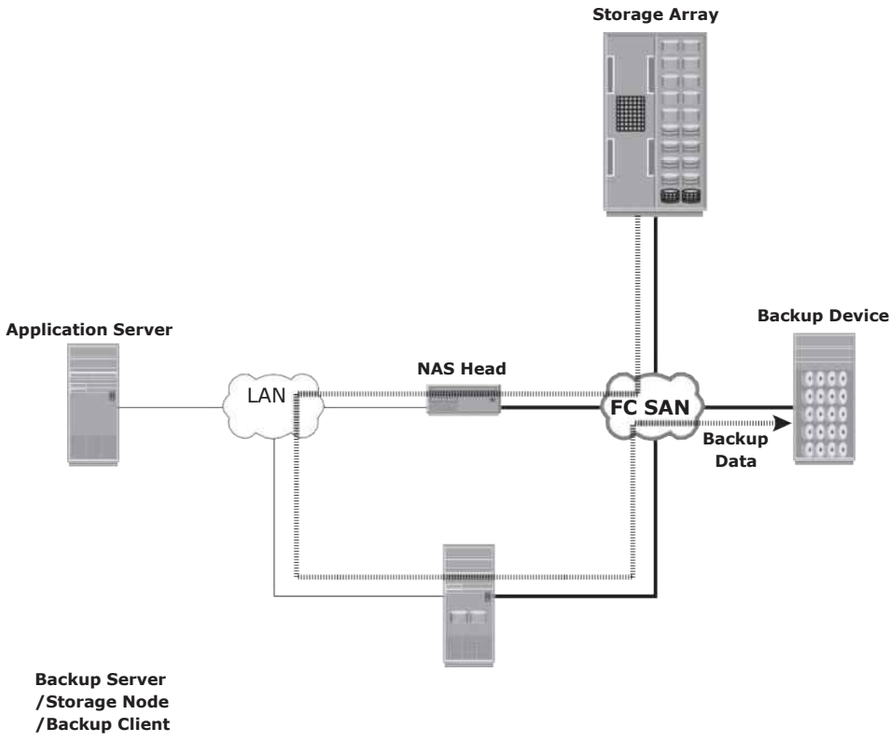
Server-Based and Serverless Backup

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



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

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



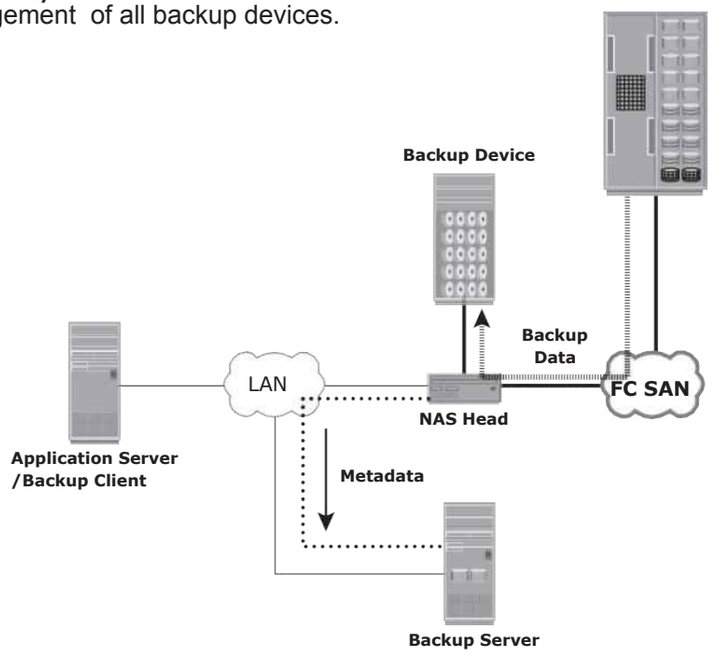
- 10-12: Serverless backup in a NAS environment

10.1.1 NDMP-Based Backup

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

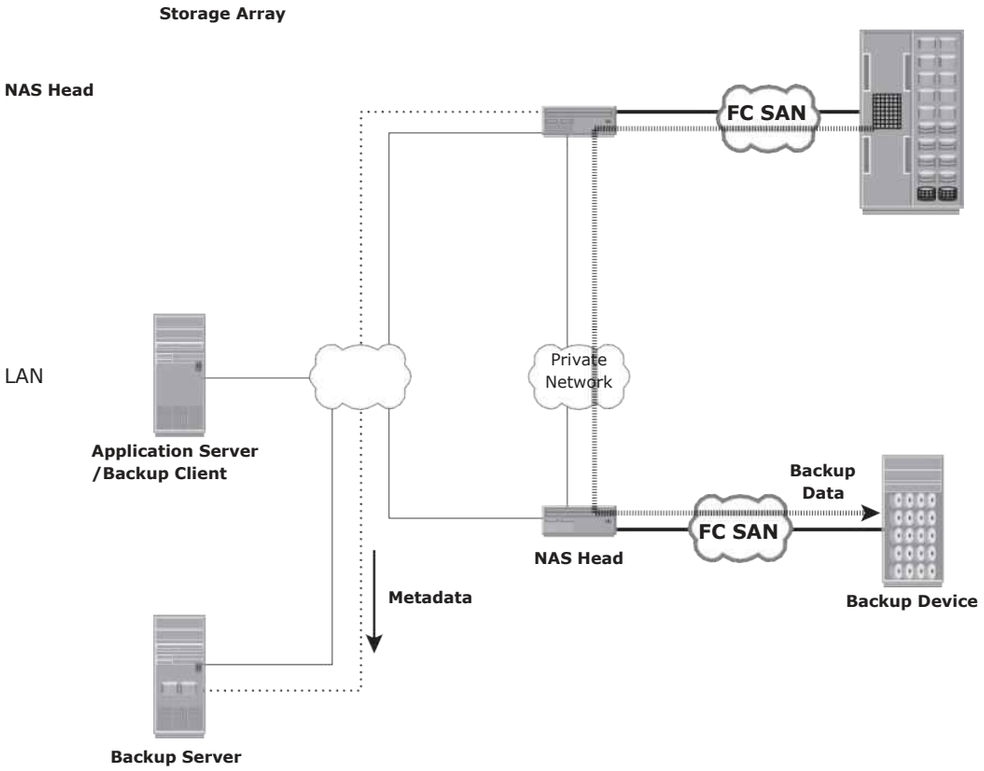
system or platform. Due to its flexibility, it is no longer necessary to transport data through the application server, which reduces the load on the application server and improves the backup speed.

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



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

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



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

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

10.2 Backup Targets

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

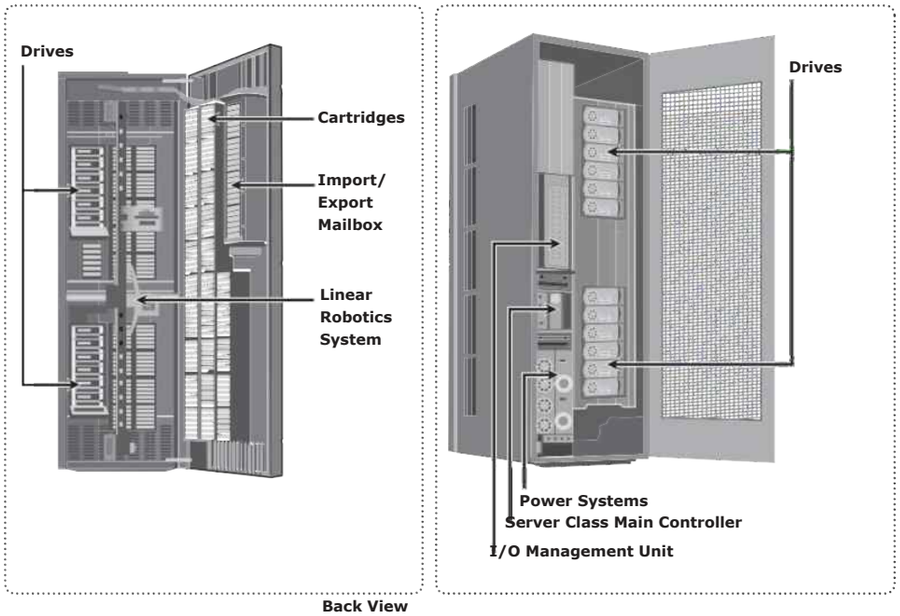
Backup to Tape

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

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

Physical Tape Library

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



- 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 - 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 buff- ering 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 - 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.

- 10-16: Multiple streams on tape media

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

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

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

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

Limitations of Tape

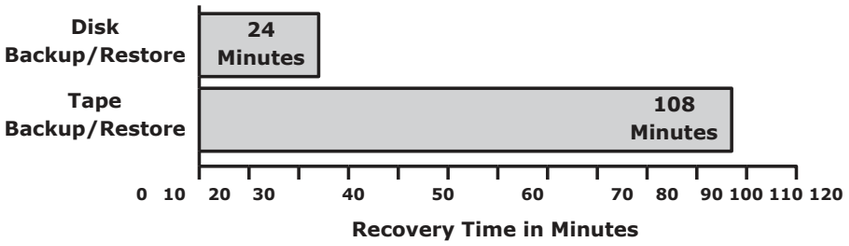
Tapes are primarily used for long-term offsite storage because of their low cost. Tapesmust be stored in locations with a controlled environment to ensure pres- ervation of the media and to prevent data corruption. Data access in a tape is sequential, which can slow backup and recovery operations. Tapes are highly susceptible to wear and tear and usually have shorter shelf life. Physical trans- portation 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 advan- tages. 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 storagesystemsoffersclear advantagesduetotheir inherent random access and RAID-protection capabilities. Inmost 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. - 10-17 illustrates a recovery scenario comparing tape versus disk in a Microsoft Exchange environment that supports 800 users with a 75 MB mailbox size and a 60 GB database. As shown in the figure, a restore from the

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



- 10-17: Tape versus disk restore

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

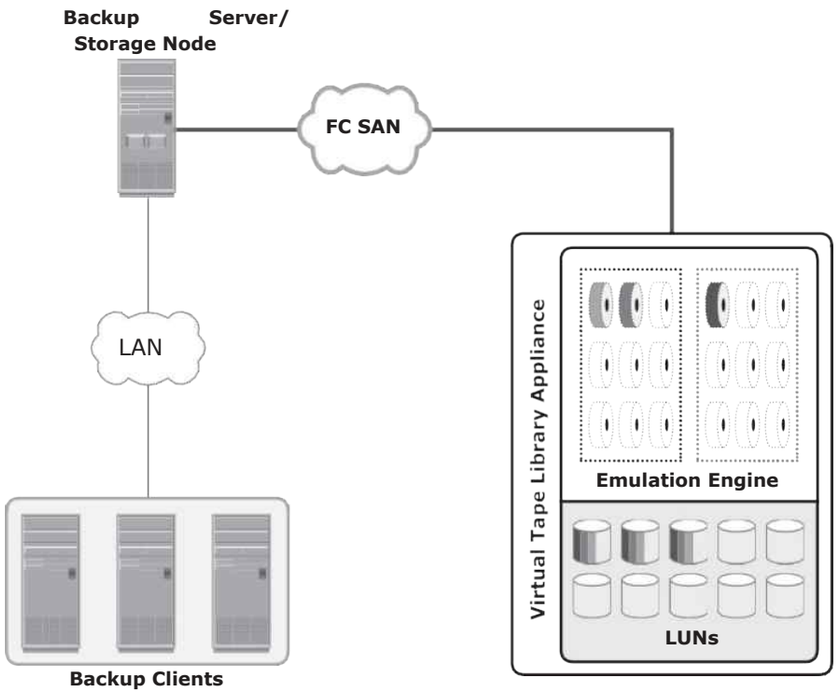
Backup to disk does not offer any inherent offsite capability and is dependent on other technologies, such as local and remote replication. In addition, some backup products require additional modules and licenses to support backup to disk, which may also require additional configuration steps, including cre- ation 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. - 10-18 shows a virtual tape library. Virtual tape libraries use disks as backup media. Emulation software has a database with a list of virtual tapes, and each virtual tape is assigned space on a LUN. A virtual tape can span multiple LUNs if required. File system aware- ness is not required while backing up because the virtual tape solution typically uses raw devices.



- 10-18: Virtual tape library

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

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

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

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

Data replication over IP is available with most of the virtual tape library appli- ances. 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 instal- lation 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 produc- tion and backup data.

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

Table 10-1: Backup Targets Comparison

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

Data Deduplication for Backup

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

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

Data Deduplication Methods

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

File-level deduplication (also called *single-instance storage*) detects and removes redundant copies of identical files. It enables storing only one copy of the file; the subsequent copies are replaced with a pointer that points to the original file.

File-level deduplication is simple and fast but does not address the problem of duplicate content inside the files. For example, two 10-MB PowerPoint presenta-

tions with a difference in just the title page are not considered as duplicate files,

and each file will be stored separately.

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

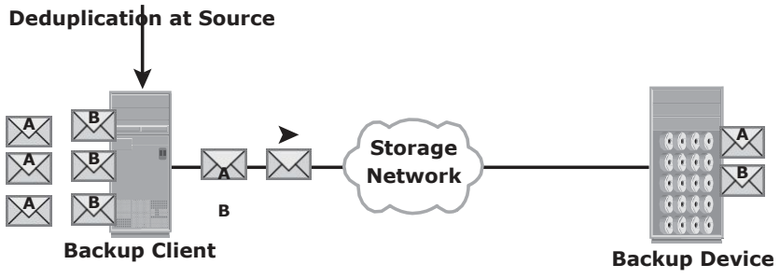
boundary for only that segment is adjusted, leaving the remaining segments unchanged. This method vastly improves the ability to find duplicate data segments compared to fixed-block.

Data Deduplication Implementation

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

Source-Based Data Deduplication

Source-based data deduplication eliminates redundant data at the source before it transmits to the backup device. Source-based data deduplication can dramati- cally 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 networkbandwidth. There is also a substantial reductioninthe capacity required to store the backupimages. - 10-19 showssource-based data deduplication.



- 10-19: Source-based data deduplication

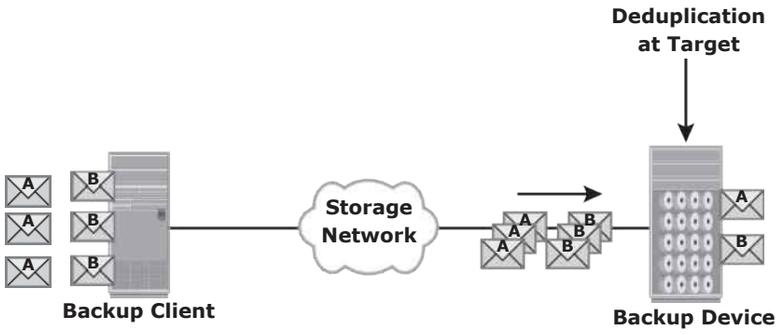
Source-based deduplicationincreases the overheadon the backupclient, which impacts the performance of the backup and application running on the client. Source-based deduplication might also require a change of backup software if it is not supported by backup software.

Target-Based Data Deduplication

Target-based data deduplication is an alternative to source-based datadeduplication. Target-based data deduplication occurs at the backup device, which offloads the backup client from the deduplication process. - 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 deduplicationoccurs 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.



- 10-20: Target-based data deduplication

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

Post-process deduplication enables the backup data to be stored or written on the backup device first and then deduplicated later. This method is suitable for situations with tighter backup windows. However, post-process deduplica-

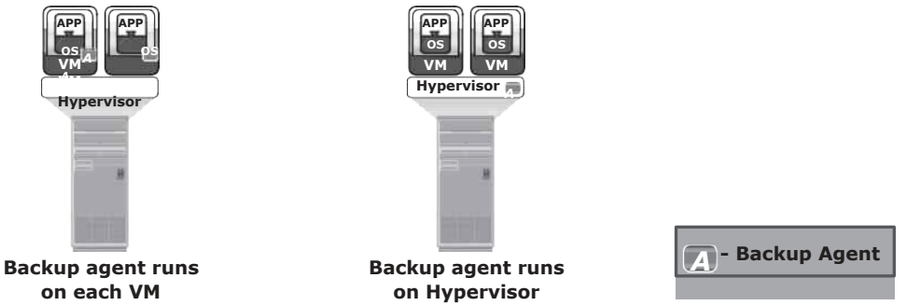
tion requires more storage capacity to store the backup images before they are deduplicated.

Backup in Virtualized Environments

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

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

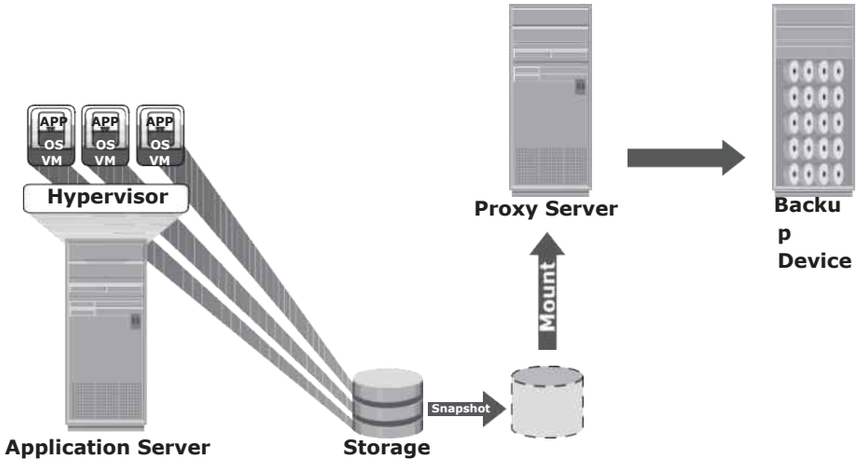
If the backup agent is installed on the hypervisor, the VMs appear as a set of files to the agent. So, VM files can be backed up by performing a file system backup from a hypervisor. This approach is relatively simple because it requires having the agent just on the hypervisor instead of all the VMs.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



- 10-21: Traditional VM backup

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

Image-based backup operates at the hypervisor level and essentially takes a snap- shot 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 applicationconfigurations. The backup is saved as a single file called an —image,ll and this image is mounted on the separate physical machine–proxy server, which acts as a backup client. The backup software then backs up these image files normally. (see - 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.

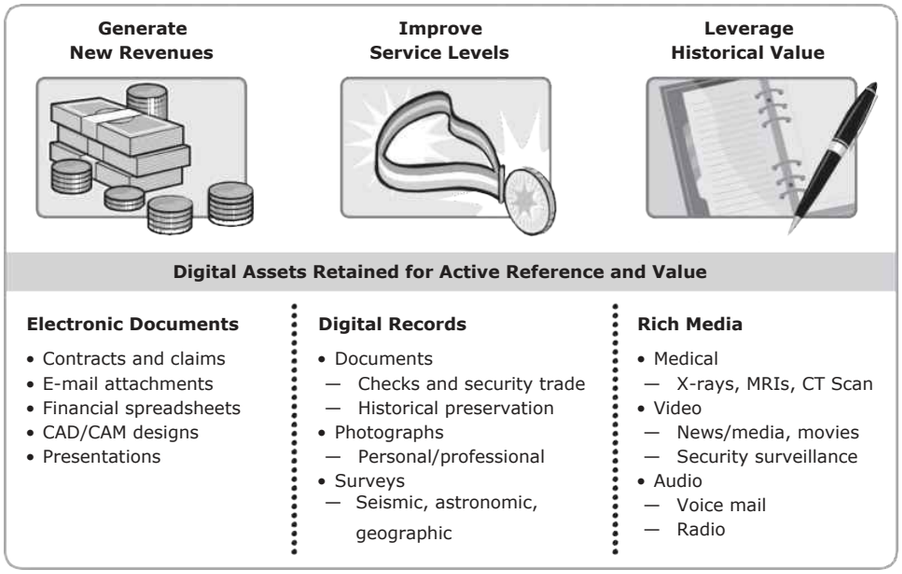


- 10-22: Image-based backup

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

Data Archive

In the life cycle of information, data is actively created, accessed, and changed. As data ages, it is less likely to be changed and eventually becomes —fixedǁ 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. - 10-23 shows some examples of fixed content.



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

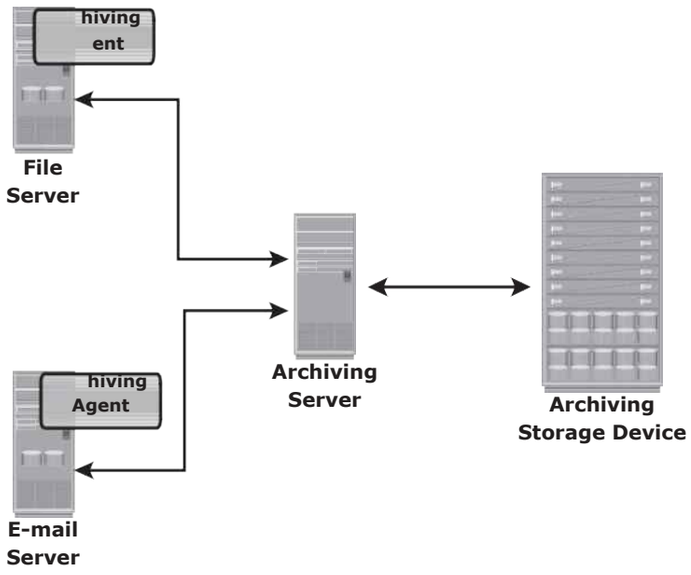
Content addressed storage (CAS) is disk-based storage that has emerged as an alternative to tape and optical solutions. CAS meets the demand to improve data accessibility and to protect, dispose of, and ensure service-level agreements (SLAs) for archive data. CAS is detailed in Chapter 8.

Archiving Solution Architecture

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

An archiving agent is software installed on the application server. The agent is responsible forscanning the datathat can be archivedbased on the policydefined on the archiving server. After the data is identified for archiving, the agent sends the data to the archiving server. Then the original data on the application server

is replaced with a stub file. The stub file contains the address of the archived data. The size of this file is small and significantly saves space on primary stor- age. This stub file is used to retrieve the file from the archive storage device.



- 10-24: Archiving solution architecture

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

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

Use Case: E-mail Archiving

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

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

Use Case: File Archiving

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

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

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

EMC NetWorker

The EMC NetWorker backup and recovery software centralizes, automates, and acceleratesdatabackup and recoveryoperations across the enterprise. Following are the features of EMC NetWorker:

- Supports heterogeneous platforms, suchas Windows, UNIX, and Linux, and also supports virtualenvironments
- Supports clustering technologies and open-file backup
- Supports different backup targets: tapes, disks, and virtual tapes
- Supports Multiplexing (or multistreaming) of data
- Provides both source-based and target-based deduplication capabilities byintegrating with EMC Avamarand EMC Data Domainrespectively
- Uses 256-bit AES(advancedencryptionstandard) encryptionto provide security for the backup data. NetWorker hosts are authenticated using strongauthenticationbasedonthe Secure Sockets Layer(SSL) protocol.
- The cloud-backup option in NetWorker enables backing up data to both private and public cloudconfigurations.

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

EMC Avamar

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

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

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

The features of EMC Avamar follows:

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

EMC Data Domain

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

Data Domain deduplication storage systems provide the following unique advantages:

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

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

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

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

Replication Terminology

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

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

Uses of Local Replicas

One or more local replicas of the source datamaybecreatedforvarious purposes, including the following:

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

operations and servicing data for backup operations. The local replica contains an exact point-in-time (PIT) copy of the source data, and there- fore 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 produc- tion 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 orupgrades. Forexample, anorganizationmayusethereplicatotestthe production application upgrade; if the test is successful, the upgrade may be implemented on the productionenvironment.
- **Datamigration:**Anotheruseforalocalreplicaisdatamigration.Datamigra- tionsare performed forvariousreasons, suchasmigrating fromasmaller capacity LUNto one of alargercapacityfornewerversions of the application.

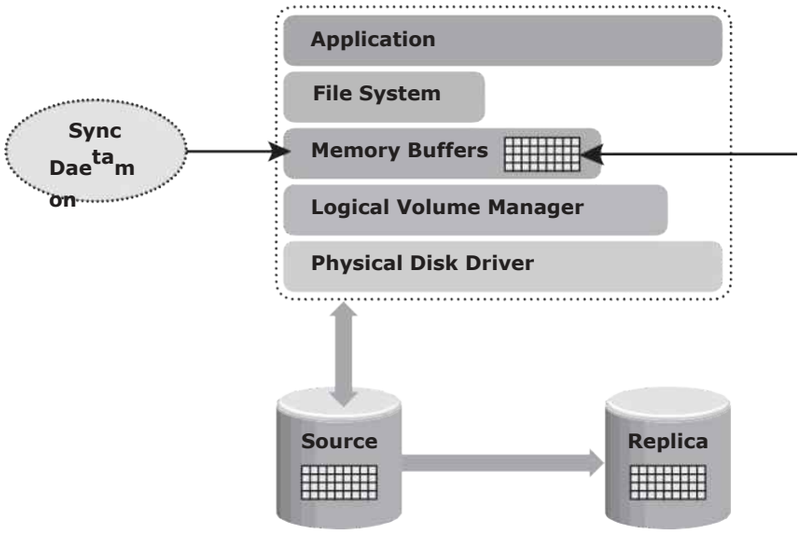
Replica Consistency

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

Consistency of a Replicated File System

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

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



- 11-1: Flushing the file system buffer

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

Consistency of a Replicated Database

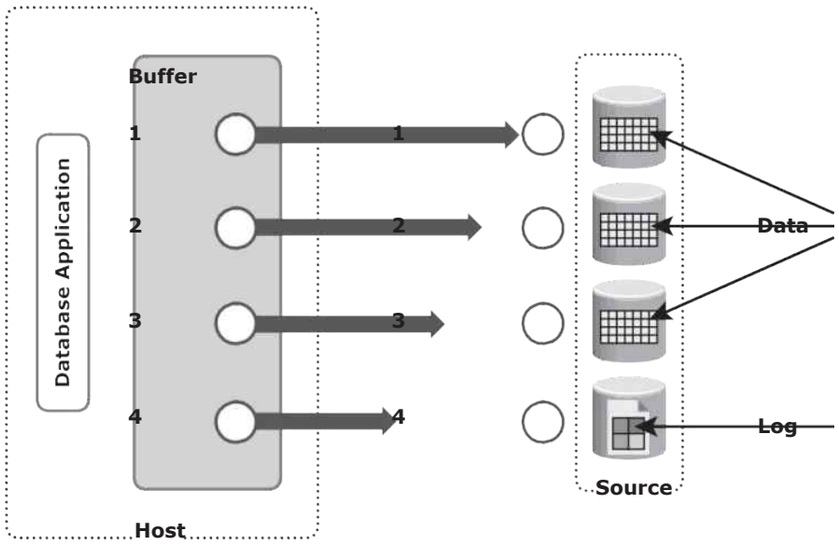
A database may be spread over numerous files, file systems, and devices. All of these must be replicated consistently to ensure that the replica is restorable and restartable. Replication is performed with the database offline or online. If the database is offline during the creation of the replica,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

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

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

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

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

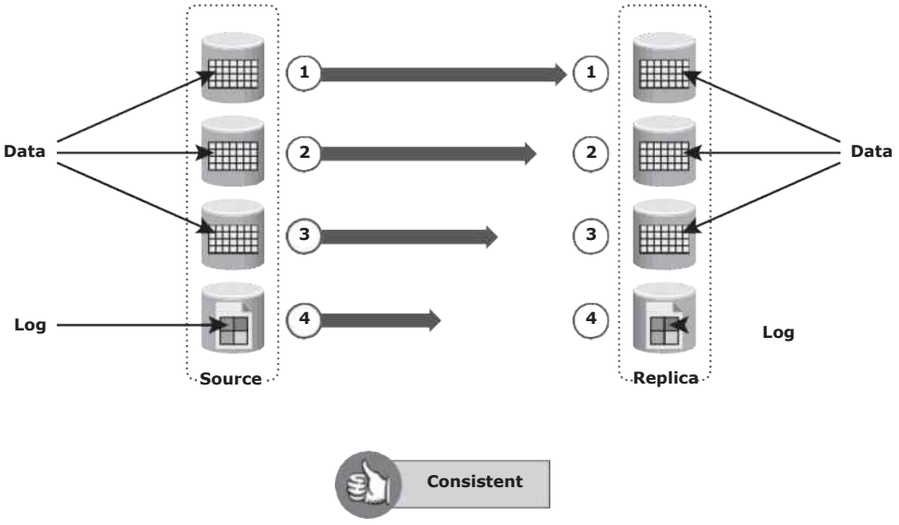


- 11-2: Dependent write consistency on sources

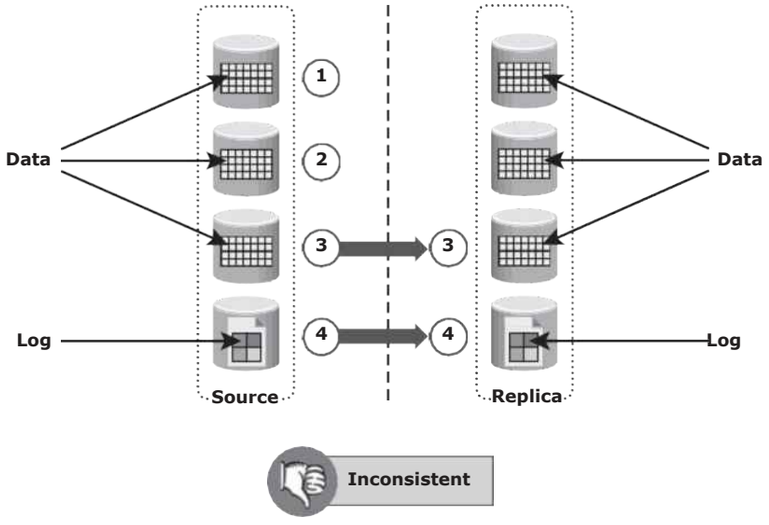
When the replica is created, all the writes to the source devices must be captured on the replica devices to ensure data consistency. - 11-3 illustrates the process of replication from the source to the replica. I/O transactions 1 to 4 must be carried out for the data to be consistent on the replica.

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

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



- 11-3: Dependent write consistency on replica



- 11-4: Inconsistent database replica

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

Local Replication Technologies

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

Host-Based Local Replication

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

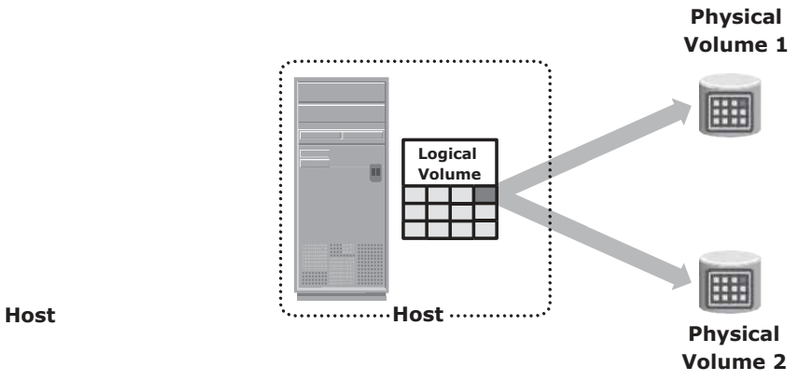
LVM-Based Replication

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

In LVM-based replication, each *logical block* in a logical volume is mapped to two physical blocks on two different physical volumes, as shown in - 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 inde- pendently accessed.

Advantages of LVM-Based Replication

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



- 11-5: LVM-based mirroring

Limitations of LVM-Based Replication

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

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

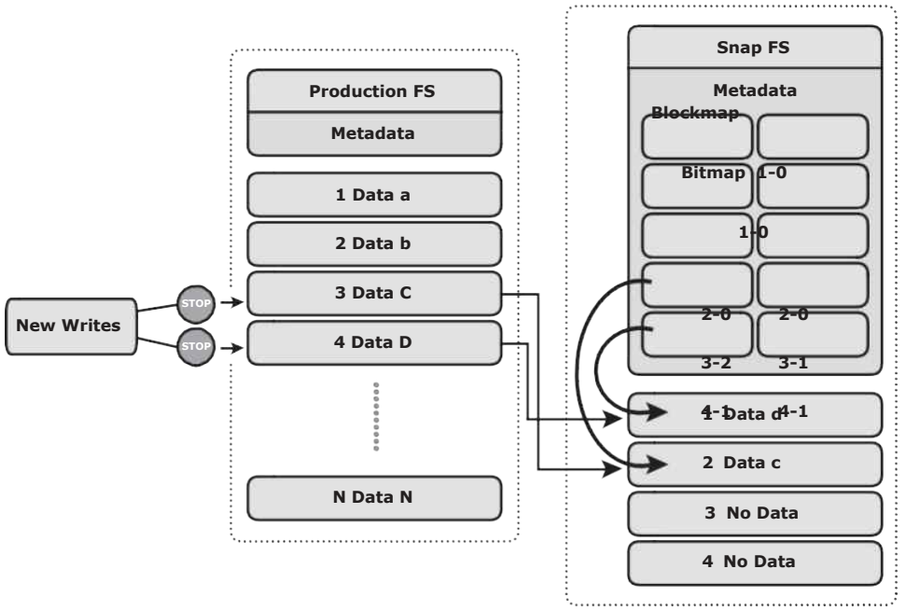
File System Snapshot

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

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

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

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

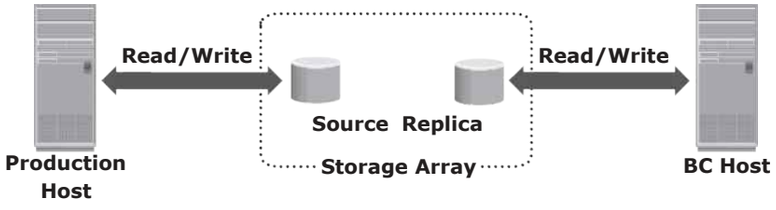


- 11-6: Write to production FS

Storage Array-Based Local Replication

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

Inthisreplication, the required number of replicadevicesshouldbe selected on thesame arrayand thendatashouldbe replicatedbetweenthesource-replicapairs. - 11-7 shows a storage array-based local replication, where the source and target are in the same array and accessed by different hosts.

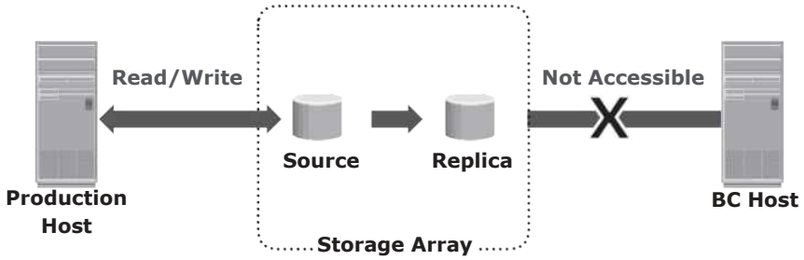


- 11-7: Storage array-based local replication

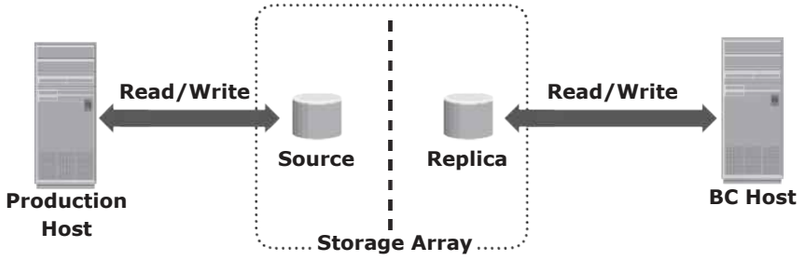
Storage array-based local replication is commonly implemented in three ways: full-volumemirroring, pointer-based full-volumereplication, and pointer-based virtual replication. Replica devices are also referred as target devices, accessible by other hosts.

Full-Volume Mirroring

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



(a) Full Volume Mirroring with Source Attached to Replica



(b) Full Volume Mirroring with Source Detached from Replica

- 11-8: Full-volume mirroring

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

After the synchronization is complete, the target can be detached from the source and made available for other business operations. - 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 pro- duction and business continuity hosts respectively.

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

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

Pointer-Based, Full-Volume Replication

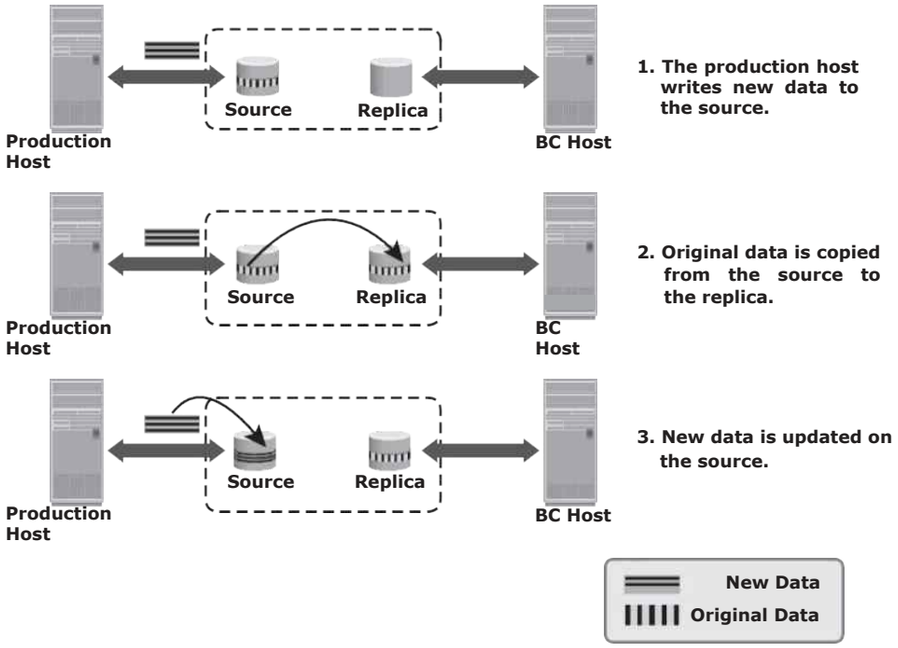
Another method of array-based local replication is *pointer-based full-volume replication*. Similar to full-volume mirroring, this technology can provide full copies of the source data on the targets. Unlike full-volume mirroring, the target is immediately accessible by the BC host after the replication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

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

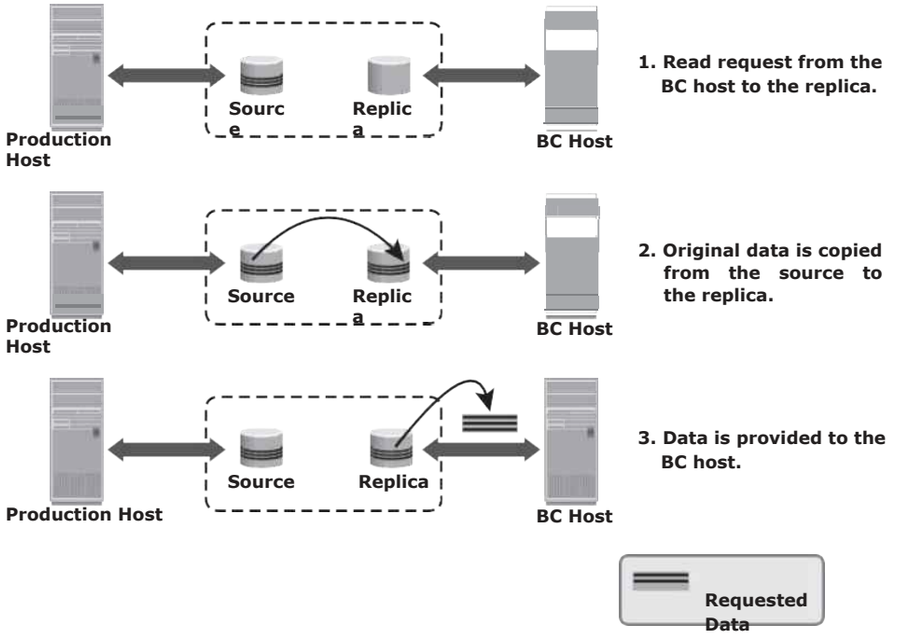
In CoFA, after the replication session is initiated, the data is copied from the source to the target only when the following condition occurs:

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

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

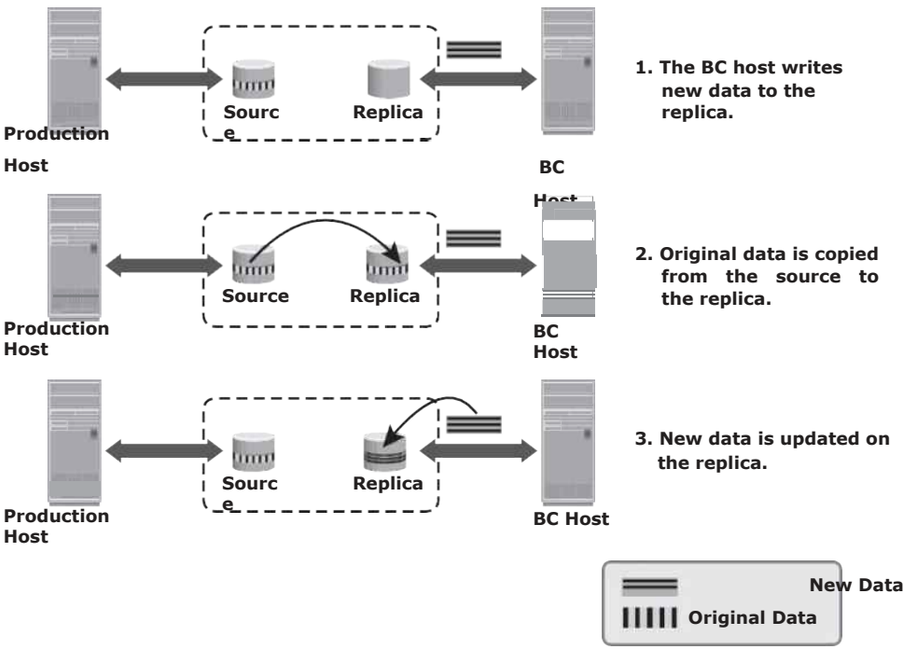


- 11-9: Copy on first access (CoFA) — write to source



- 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 - 11-11).



- 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.ll

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-volumereplicationtechnologies require the target devicestobe at least as large as the source devices. In addition, full-volume mirroring and pointer-
based full-volume replication in the Full Copy mode can provide incremental resynchronization and restore capabilities.

Pointer-Based Virtual Replication

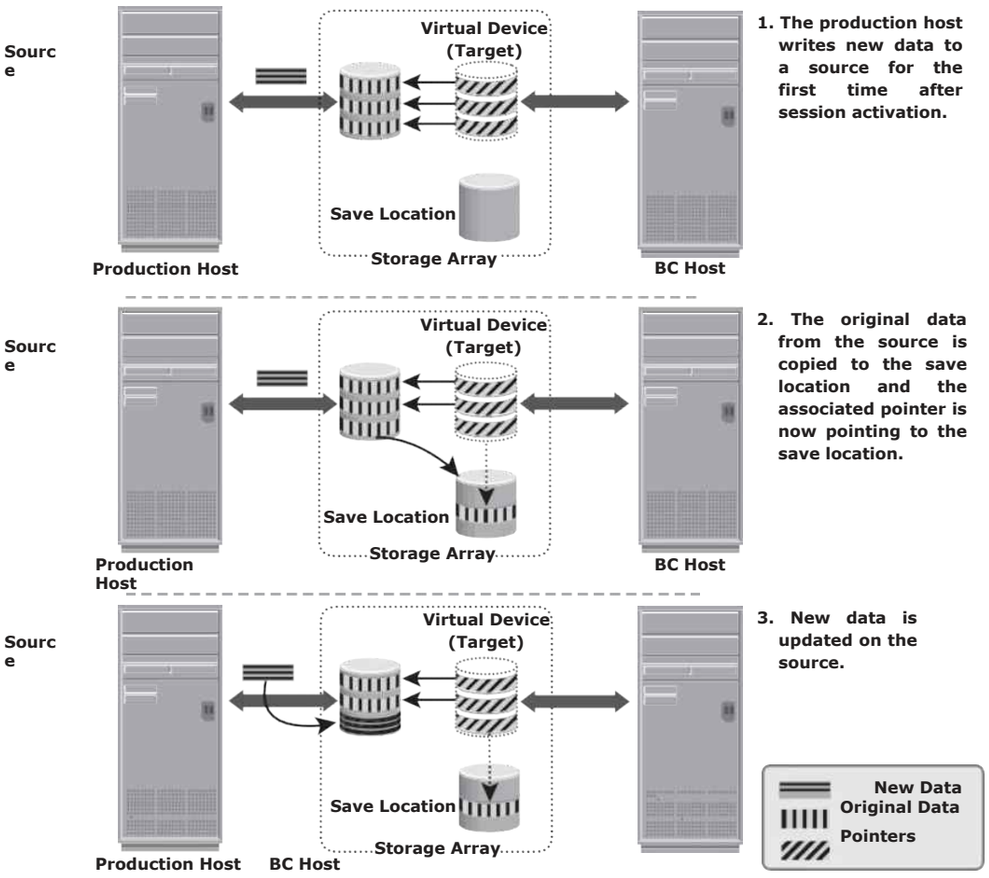
In *pointer-based virtual replication*, at the time of the replication session activation, the target contains pointers to the location of the data on the source. The target does not contain data at any time. Therefore, the target is known as a *virtual replica*. Similar to pointer-based full-volume replication, the target is immedi- ately 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 - 11-12.

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

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

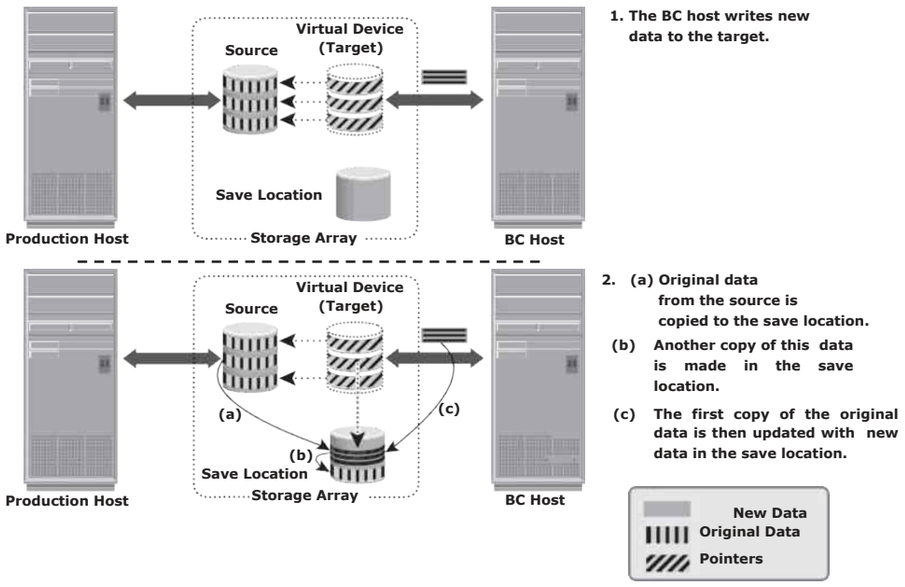
Data on the target is a combined view of unchanged data on the source and data on the save location. Unavailability of the source device invalidates the data on the target. The target contains only pointers to the data, and therefore, the physical capacity required for the target is a fraction of the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



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

Network-Based Local Replication

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



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

Continuous Data Protection

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

In CDP, data changes are continuously captured and stored in a separate location from the primary storage. Moreover, RPOs are random and do not need to be defined in advance. With CDP, recovery from data corruption poses no

problem because it allows going back to a PIT image prior to the data corruption incident. CDP uses a *journal volume* to store all data changes on the primary

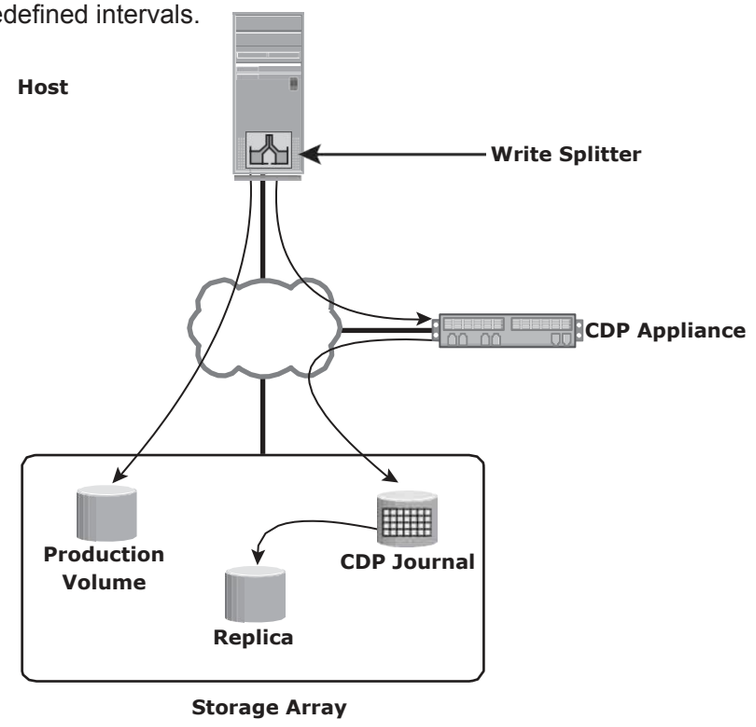
storage. The journal volume contains all the data that has changed from the time the replication session started. The amount of space that is configured for the journal determines how far back the recovery points can go. CDP is

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

CDP appliance is an intelligent hardware platformthat 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

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



- 11-14: Continuous data protection — local replication

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

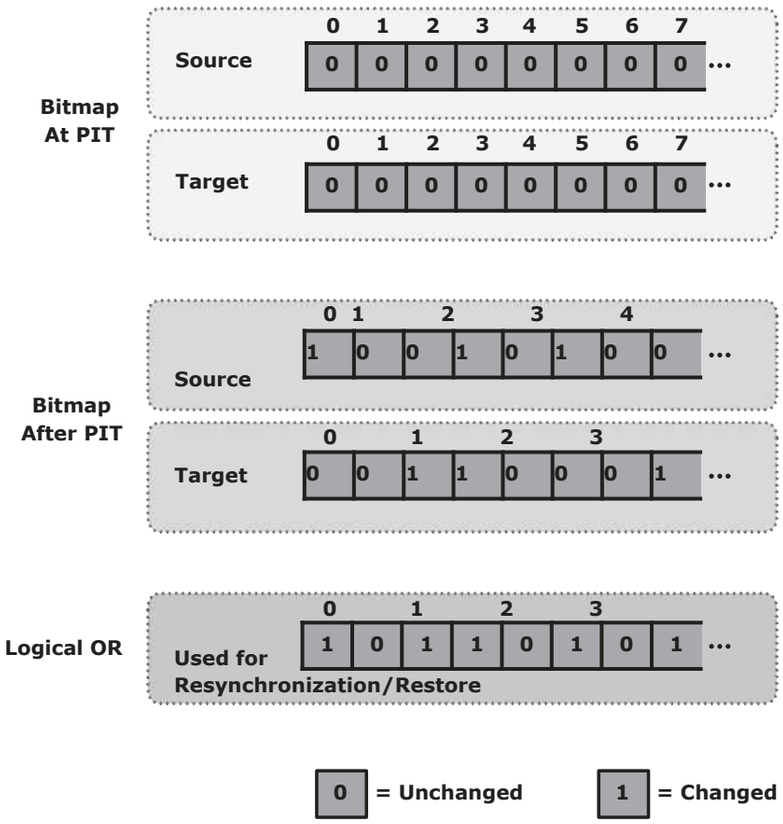
Tracking Changes to Source and Replica

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

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

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

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



- 11-15: Tracking changes

Restore and Restart Considerations

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

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

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

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

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

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

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

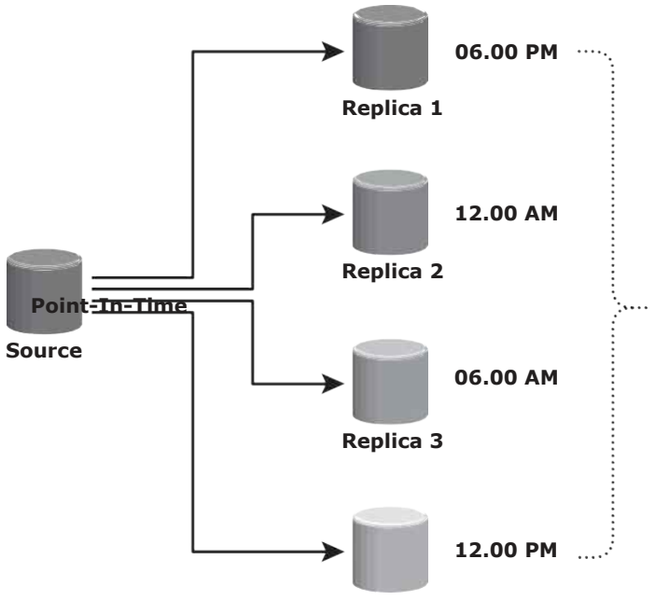
Table 11-1: Comparison of Local Replication Technologies

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

Creating Multiple Replicas

Moststoragearray-basedreplicationtechnologiesenablesourcedevices to maintain replicationrelationshipswithmultipletargets. Changesmade tothe sourceandeach of the targets can be tracked. This enables incremental resynchronization of the targets. Each PIT copycan be usedfordifferent BC activitiesand as a restore point.

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



- 11-16: Multiple replicas created at different PIT

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

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

activities can be performed using the other replicas.

Local Replication in a Virtualized Environment

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

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

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

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

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

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

EMC TimeFinder

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

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

TimeFinder/Clone

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

TimeFinder/Snap

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

EMC SnapView

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

SnapView Snapshot

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

- **SnapView session:** The SnapView snapshot mechanism is activated when asession starts and deactivated whenasessionstops. A snapshot appears—offlinell untilthereisan active session. Multiplesnapshotscan be included in a session.

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

SnapView Clone

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

EMC RecoverPoint

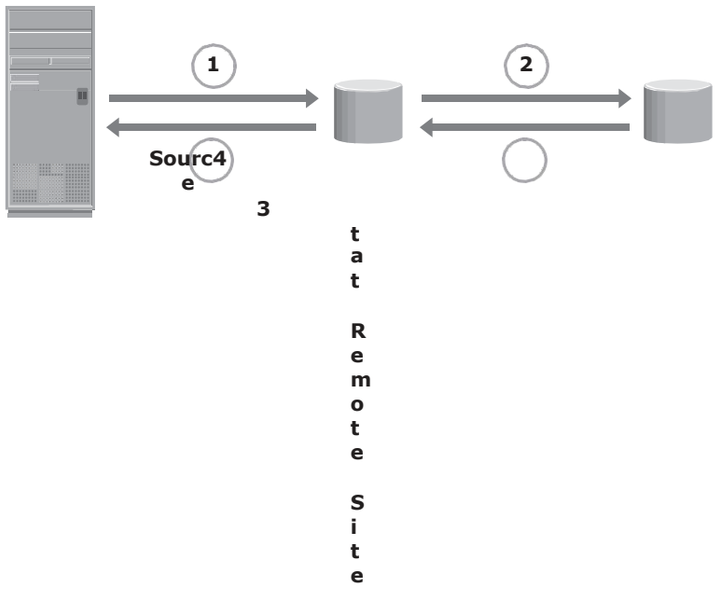
RecoverPoint is a high-performance, cost-effective, single product that provides local and remote data protection for both physical and virtual environments. It provides faster recovery and unlimited recovery points. RecoverPoint provides continuous data protection and performs replication between the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

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

Modes of Remote Replication

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

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



1 The host writes data to the source.

2 Data from the source is replicated to the target at a remote site.

3 The target acknowledges back to the source.

4 The source acknowledges write complete to the host.

- 12-1: Synchronous replication

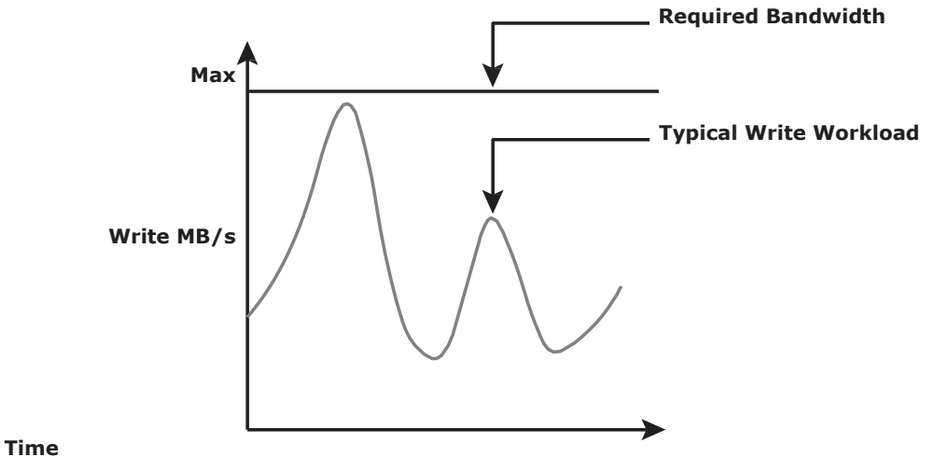
However, application response time is increased with synchronous remote replication because writes must be committed on both the source and target before sending the —write completell acknowledgment to the host. The degree of impact on response time depends primarily on the distance between sites, band-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

the two sites.

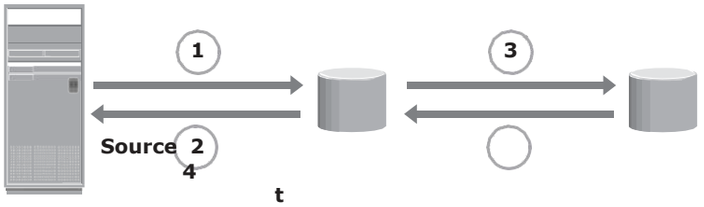
In *asynchronous remote replication*, a write is committed to the source and immediately acknowledged to the host. In this mode, data is buffered at the source and transmitted to the remote site later (see - 12-3).

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

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



- 12-2: Bandwidth requirement for synchronous replication



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

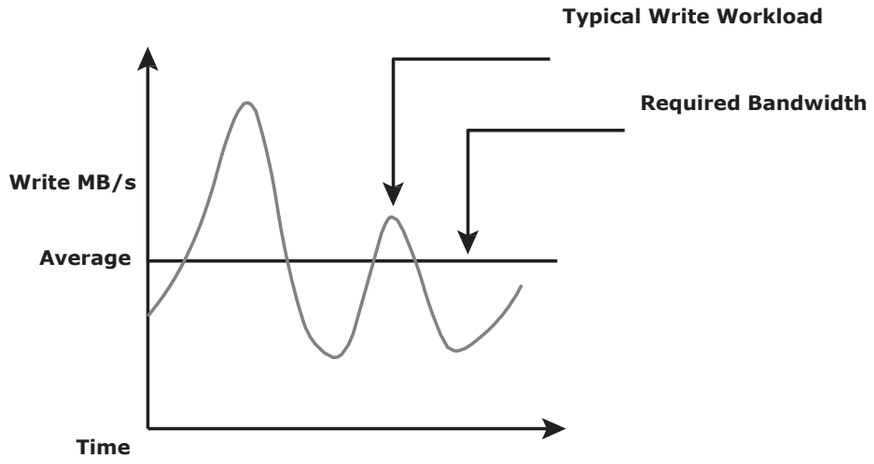
The wri2te is immediately acknowledged to the host.

3 **Data is transmitted to the target at a remote site later.**

The target a4cknowledges back to the source.

- 12-3: Asynchronous replication

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



- 12-4: Bandwidth requirement for asynchronous replication

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

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

Remote Replication Technologies

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

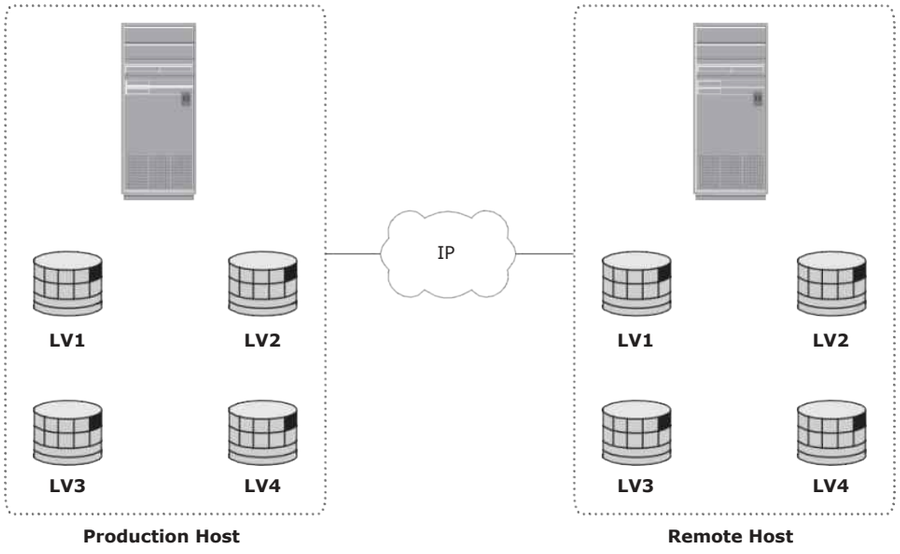
Host-Based Remote Replication

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

LVM-Based Remote Replication

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

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



- 12-5: LVM-based remote replication

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

LVM-based remote replication is independent of the storage arrays and there- foresupports replication between heterogeneous storage arrays. Mostoperating

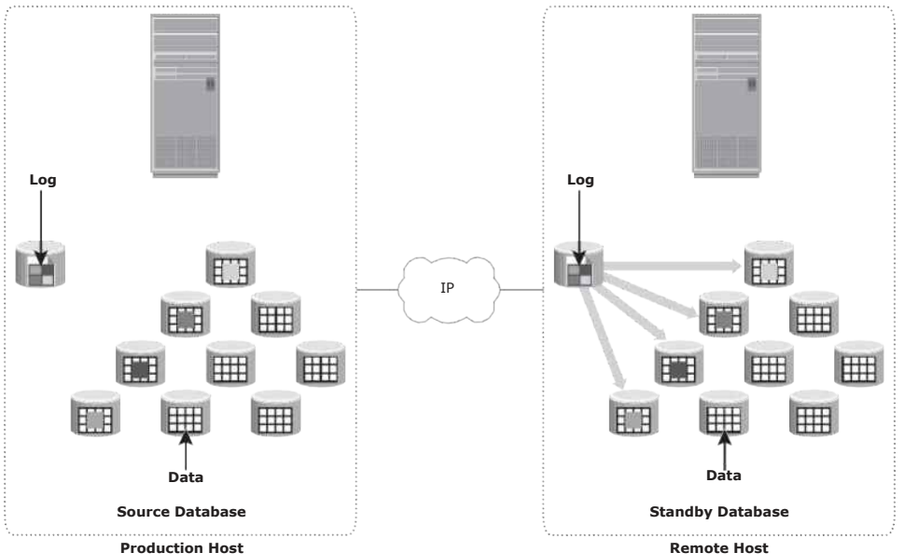
systems are shipped with LVMs, so additionallicenses and specialized hardware are not typically required.

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 - 12-6). The remote host receives the logs and applies them to the remote database.



- 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-basedremotereplication, 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.

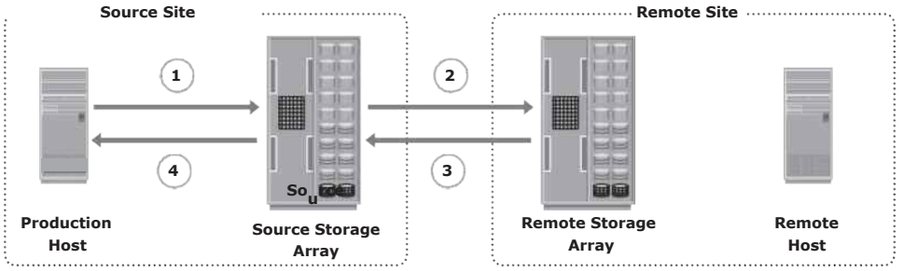
Replicationbetween arrays maybe performed insynchronous, asynchronous, or disk-buffered modes.

Synchronous Replication Mode

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

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

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



1 Write from the production host is received by the source storage array.

2 Write is then transmitted to the remote storage array.

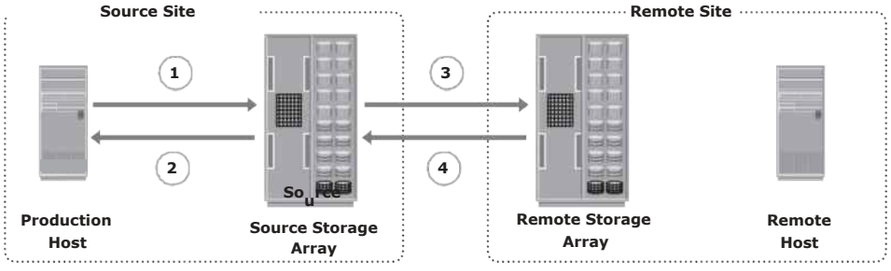
3 Acknowledgment is sent to the source storage array by the remote storage array.

4 Source storage array signals write-completion to the production host.

- 12-7: Array-based synchronous remote replication

Asynchronous Replication Mode

In array-based *asynchronous remote replication mode*, as shown in - 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.

- 12-8: Array-based asynchronous remote replication

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

Some implementations of asynchronous remote replication maintain write ordering. A timestamp and sequence number are attached to each write when it is received by the source. Writes are then transmitted to the remote array, where

they are committed to the remote replica in the exact order in which they were buffered at the source. This implicitly guarantees consistency of data on the remote replicas. Other implementations ensure consistency by leveraging the dependent write principle inherent in most DBMSs. In asynchronous remote replication, the writes are buffered for a predefined period of time. At the end of this duration, the buffer is closed, and a new buffer is opened for subsequent writes. All writes in the closed buffer are transmitted together and committed to the remote replica.

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

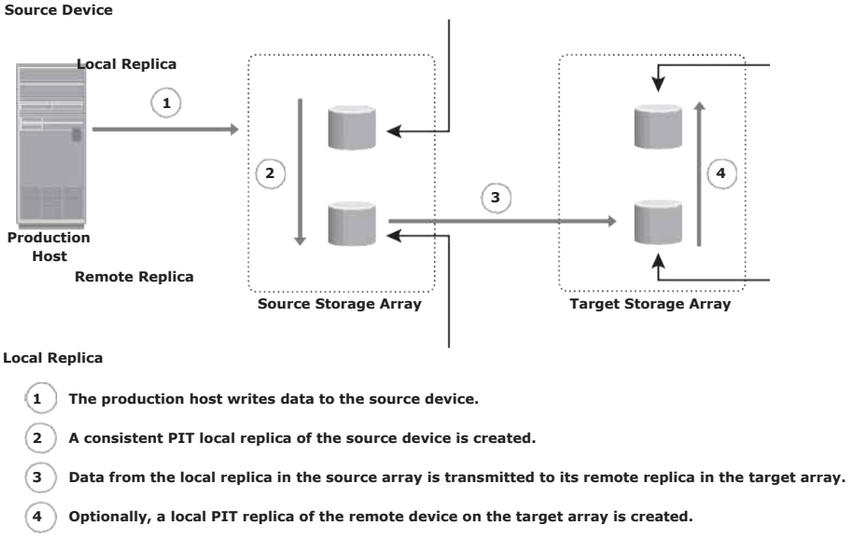
Disk-Buffered Replication Mode

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

- 12-9 showsthe sequence of operationsin adisk-bufferedremotereplication. 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 synchroni- zation of this pair, the network link is suspended, and the next local replica of the source is created. Optionally, a local PIT replica of the remote device on the target array can be created. The frequency of this cycle of operations depends on the available link bandwidth and the data change rate on the source device. Because disk-buffered technology uses local replication, changes made to the source and its replica are possible to track. Therefore, all the resynchronization operations between the source and target can be done incrementally. When compared to synchronous and asynchronous replications, disk-buffered remote replication requires less bandwidth.

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

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



- 12-9: Disk-buffered remote replication

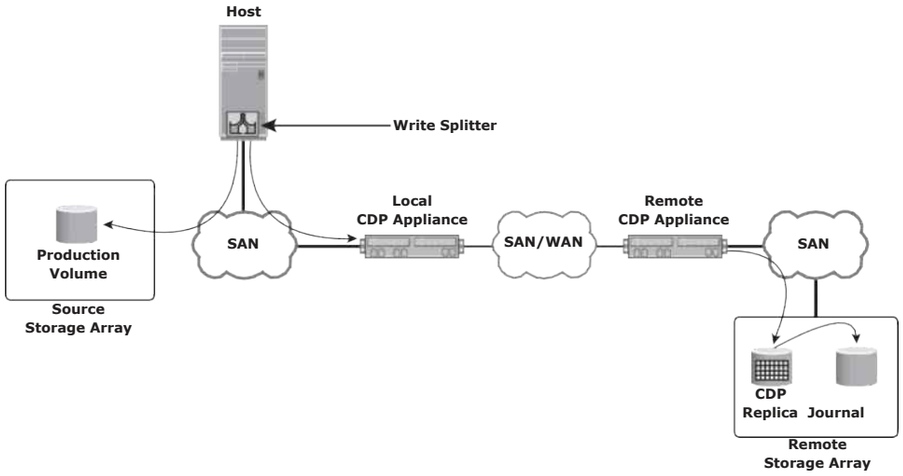
Network-Based Remote Replication

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

CDP Remote Replication

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

- 12-10 describes CDP remote replication. In this method, the replicais 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.



- 12-10: CDP remote replication

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

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

Three-Site Replication

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

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

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

remote site, providing a finite RPO. Three-site remote replication can be implemented as a cascade/multihop or a triangle/multitarget solution.

Three-Site Replication — Cascade/Multihop

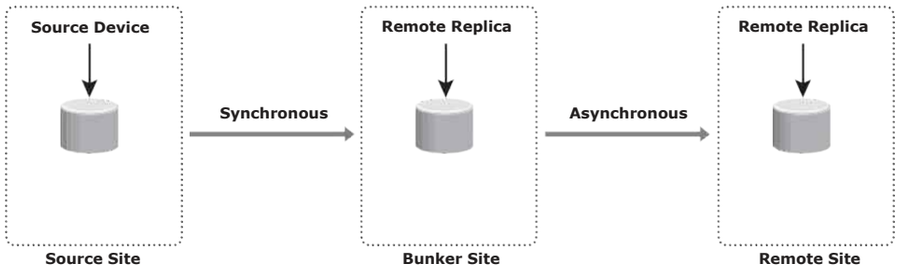
In the *cascade/multihop* three-site replication, data flows from the source to the intermediate storage array, known as a *bunker*, in the first hop, and then from a bunker to a storage array at a remote site in the second hop. Replication between the source and the remote sites can be performed in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

Synchronous + Asynchronous

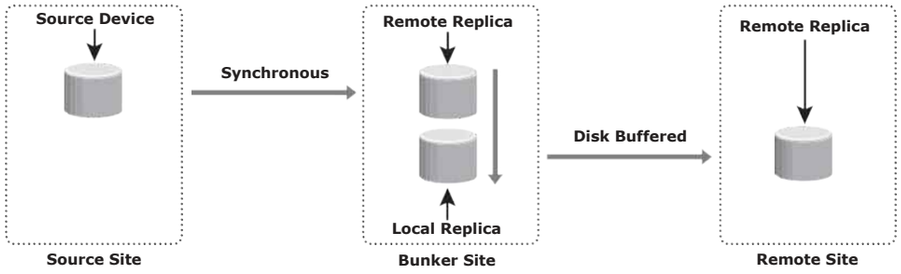
This method employs a combination of synchronous and asynchronous remote replication technologies. Synchronous replication occurs between the source and the bunker. Asynchronous replication occurs between the bunker and the remote site. The remote replica in the bunker acts as the source for asynchronous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

RPO at the remote site is usually in the order of minutes for this implementa- tion. 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 sitewith zero or near-zero dataloss. But unlike the synchronous two-site situation, there is still remote protection at the third site. The RPO between the bunker and third site could be in the order of minutes.



(a) Synchronous + Asynchronous



(b) Synchronous + Disk Buffered

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

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

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

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

Synchronous + Disk Buffered

This method employs a combination of local and remote replication technologies. Synchronous replication occurs between the source and the bunker: a consistent PIT local replica is created at the bunker. Data is transmitted from the local replica at the bunker to the remote replica at the remote site. Optionally, a local replica can be created at the remote site after data is received from the bunker. - 12-11 (b) illustrates the synchronous + disk buffered method.

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

The process to create the consistent PIT copy at the bunker and incrementally updating the remote replica occurs continuously in a cycle.

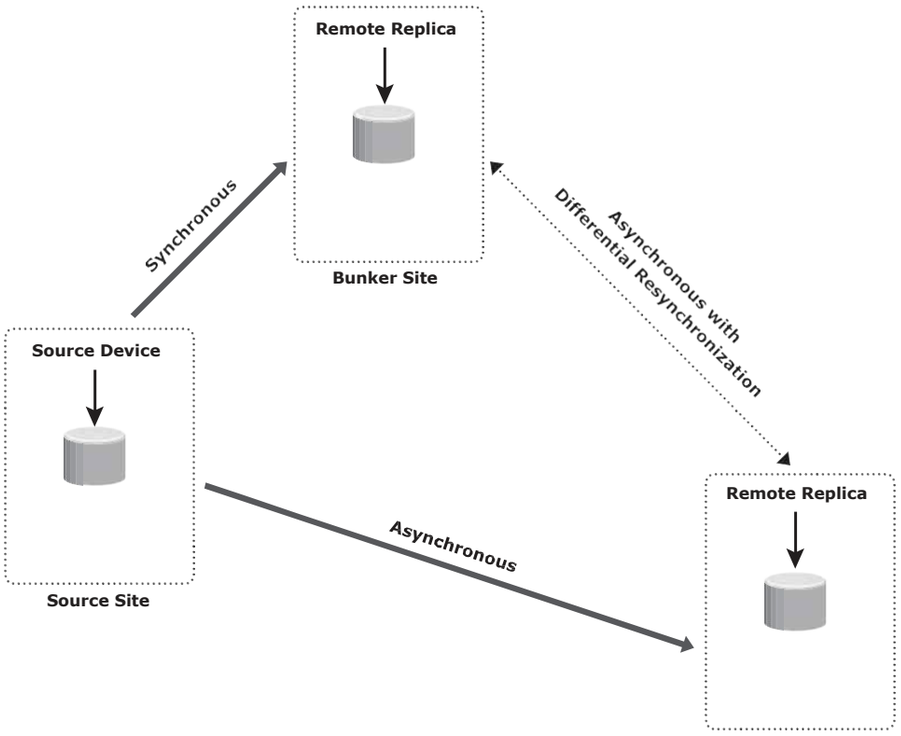
Three-Site Replication — Triangle/Multitarget

In *three-site triangle/multitarget replication*, data at the source storage array is concurrently replicated to two different arrays at two different sites, as shown in - 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.



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

A *regional disaster* in three-site triangle/multitarget replication is similar to a source site failure in two-site asynchronous replication. If failure occurs, opera- tions 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 operationcan continue uninterrupted at the source sitewhile remote disaster recovery protection is still available. A network link failure to either the source-to-bunker or the source-to-remote site does not impact production at the source site while remote disaster recovery protection is still available with the site that can be reached.

Data Migration Solutions

A data migration and mobility solution is a specialized replication technique that enables creating remote point-in-time copies. These copies can be used for data mobility, migration, content distribution, and disaster recovery. This solution

moves data between heterogeneous storage arrays. Data is moved from one array to the other over the SAN or WAN. This technology is application- and server-operating-system independent because the replication operations are performed by one of the storage arrays.

Data mobility refers to moving data between heterogeneous storage arrays for cost, performance, or any other reason. It helps implement a tiered storage strategy. *Data migration* refers to moving data from one storage array to other heterogeneous storage arrays for technology refresh, consolidation, or any other reason. The array performing the replication operations is called the *control array*. Data can be moved from/to devices in the control array to/from a remote array. The devices in the control array that are part of the replication session are called *control devices*. For every control device, there is a counterpart, a *remote device*, on the *remote array*. The terms control or remote do not indicate the direction of data flow; they indicate only the array that is performing the replication operation. The direction of data movement is determined by the replication operation.

The front-end ports of the control array must be zoned to the front-end ports of the remote array. LUN masking should be performed on the remote array to allow access to the remote devices to the front-end port of the control array. In effect, the front-end ports of the control array act as an HBA, initiating data transfer to/from the remote array.

Data migration solutions perform push and pull operations for data movement. These terms are defined from the perspective of the control array. In the *push operation*, data is moved from the control array to the remote array. The control device, therefore, acts like the source, while the remote device is the target.

In the *pull operation*, data is moved from the remote array to the control array.

The remote device is the source, and the control device is the target.

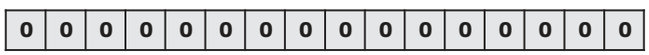
When a push or pull operation is initiated, the control array creates a protection bitmap to track the replication process. Each bit in the protection bitmap represents a data chunk on the control device. The chunk size varies with technology implementations. When the replication operation is initiated, all the bits are set to one, indicating that all the contents of the source device need to be copied to the target device. As the replication process copies data, the bits are changed to zero, indicating that a particular chunk has been copied. At the end of the replication process, all the bits become zero.

During the push and pull operations, host access to the remote device is not allowed because the control array has no control over the remote array and can- not track any change on the remote device. Data integrity cannot be guaranteed if changes are made to the remote device during the push and pull operations. The push and pull operations can be either hot or cold. These terms apply to the control devices only. In a *cold operation* the control device is inaccessible to the host during replication. Coldoperations guarantee data consistency because

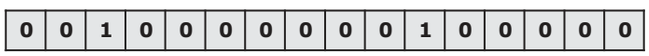
both the control and the remote devices are offline. In a *hot operation* the con- trol device is online for host operations. During hot push and pull operations, changes can be made to the control device because the control array can keep track of all changes and thus ensure data integrity.

When the hot push operationis initiated, applicationsmaybe up-and-running on the control devices. I/O to the control devices is held while the protection bitmap is created. This ensures a consistent PIT image of the data. The protec- tion bitmap is referred prior to any write to the control devices. If the bit is zero, the write is allowed. If the bit is one, the replication process holds the incoming write, copies the corresponding chunk to the remote device, and then allows the write to complete.

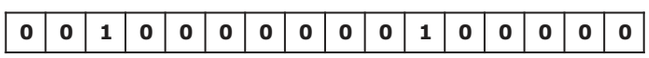
In the hot pull operation, the hosts can access control devices after starting the pull operation. The protection bitmap is referenced for every read or write operation. If the bit is zero, a read or write occurs. If the bit is one, the read or write is held, and the replication process copies the required chunk from the remote device. When the chunk is copied to the control device, the read or write is allowed to complete. The control devices are available for production soon after the pull operation is initiated and the protection bitmap is created. The control array can keep track of changes made to the control devices, so incremental push operation is possible. A second bitmap, called a *resynchronization bitmap*, is created. All the bits in the resynchronization bitmap are set to zero when a push is initiated, as shown in - 12-13 (a). As changes are made to the control device, the bits are flipped from zero to one, indicating that changes have occurred, as shown in - 12-13 (b). When resynchronization is required, the push is reinitiated and the resynchronization bitmap becomes the new protection bitmap, as shown in - 12-13 (c), and only the modified chunks are transmitted to the remote devices. An incremental pull operation is not possible because tracking changes is not performed at the remote device.



(a) Resynchronization Bitmap When Push Is Initiated



(b) Resynchronization Bitmap When Data Chunks Are Updated



(c) Resynchronization Bitmap Becomes Protection Bitmap

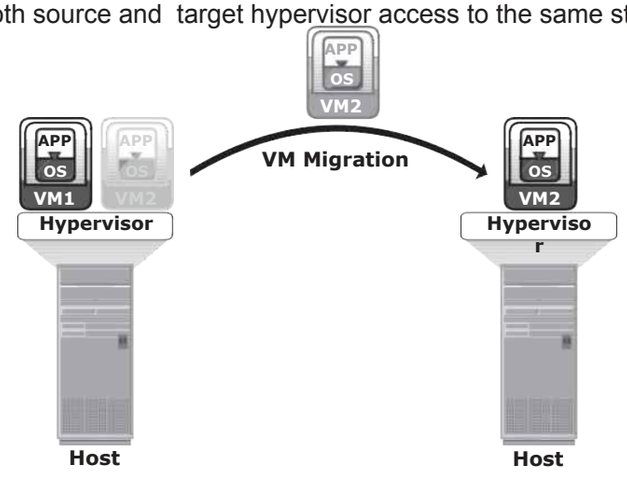
- 12-13: Bitmap status during push operation

Remote Replication and Migration in a Virtualized Environment

In a virtualized environment, all VM data and VM configuration files residing on the storage array at the primary site are replicated to the storage array at the remote site. This process remains transparent to the VMs. The LUNs are replicated between the two sites using the storage array replication technology. This replication process can be either synchronous (limited distance, near zero RPO) or asynchronous (extended distance, nonzero RPO).

Virtual machine migration is another technique used to ensure business con- tinuity in case of hypervisor failure or scheduled maintenance. VM migration is the process to move VMs from one hypervisor to another without powering off the virtual machines. VM migration also helps in load balancing when multiple virtual machines running on the same hypervisor contend for resources. Two commonly used techniques for VM migration are hypervisor-to-hypervisor and array-to-array migration.

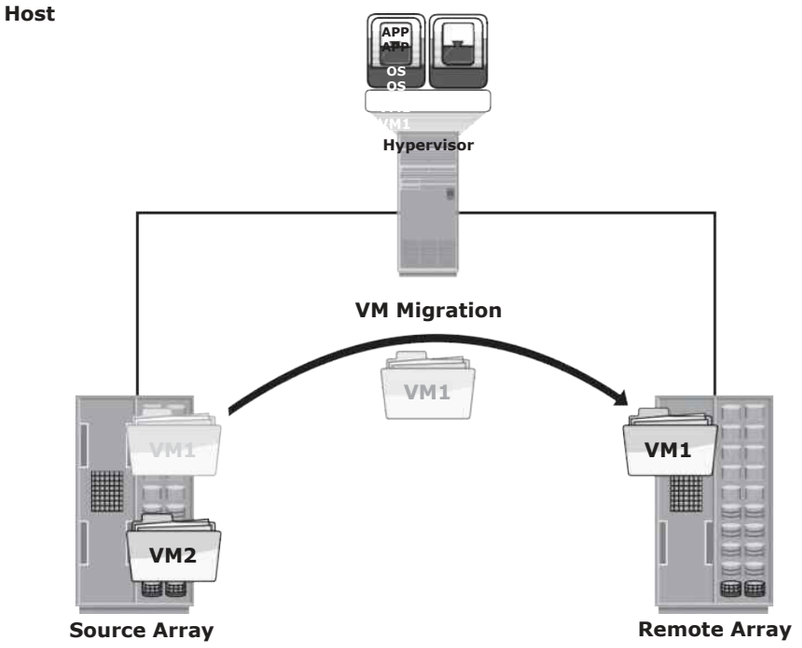
In hypervisor-to-hypervisor VM migration, the entire active state of a VM is moved from one hypervisor to another. - 12-14 shows hypervisor-to-hypervisor VM migration. This method involves copying the contents of virtual machine memory from the source hypervisor to the target and then transferring the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



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

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

VMs across dissimilar storage arrays. - 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.



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

Concepts in Practice: EMCSRDF, EMC MirrorView, and EMC RecoverPoint

This section discusses the EMC products for remote replication. EMC Symmetrix Remote Data Facility (SRDF) and EMC MirrorView are the storage array-based remote application software supported by EMC Symmetrix and VNX, respec- tively. EMC RecoverPoint is a network-based replication solution. For the latest information, visit www.emc.com

EMC SRDF

SRDF offers a family of technology solutions to implement storage array-based remote replication. The SRDF family of software includes the following:

- **SRDF/Synchronous(SRDF/S):** A remote replication solution that creates a synchronous replica at one or more Symmetrix targets located within campus, metropolitan, or regional distances. SRDF/S provides a no-data-loss solution (near zero RPO) if a local disaster occurs.
- **SRDF/Asynchronous (SRDF/A):** A remote replication solution that enables the source to asynchronously replicate data. It incorporates delta set technology, which enables write ordering by employing a buffering mechanism. SRDF/A provides minimal data loss if a regional disaster occurs.
- **SRDF/DM:** A data migration solution that enables data migration from the source to the target volume over extended distances.
- **SRDF/Automated Replication (SRDF/AR):** A remote replication solution that uses both SRDF and TimeFinder/Mirror to implement disk-buffered replication technology. It is offered as SRDF/AR Single-hop for two-site replication and SRDF/AR Multihop for three-site cascade replication. SRDF/AR provides a long distance solution with RPO in the order of hours.
- **SRDF/Star:** Three-site multitarget remote replication solution that consists of primary (production), secondary (bunker), and tertiary (remote) sites. The replication between the primary and secondary sites is synchronous, whereas the replication between the primary and tertiary sites is asynchronous. If a primary site outage occurs, EMC's SRDF/Star solution enables organizations to quickly move operations and reestablish remote replication between the remaining two sites.

EMC MirrorView

The MirrorView software enables EMC VNX storage array-based remote replication. It replicates the contents of a primary volume to a secondary volume that resides on a different VNX storage system. The MirrorView family consists of MirrorView/Synchronous (MirrorView/S) and MirrorView/Asynchronous (MirrorView/A) solutions.

EMC RecoverPoint

EMC RecoverPoint Continuous Remote Replication (CRR) is a comprehensive data protection solution that provides bidirectional synchronous and asynchronous replication. In normal operations, RecoverPoint CRR enables users to

recover data remotely to any point in time. RecoverPoint dynamically switches between synchronous and asynchronous replication based on the policy for performance and latency.