IBM FlashSystem A9000R
Version 12.2.1

Product Overview

SC27-8558-11

Note
Before using this document and the product it supports, read the information in "Notices" on page 133.

Edition notice

Publication number: SC27-8558-11. This publication applies to IBM FlashSystem A9000R version 12.2.1 and to all subsequent releases and modifications until otherwise indicated in a newer publication.

Contents

Figures

Tables

About this document

This document provides a technical overview of the IBM FlashSystem® A9000R functional features and capabilities.

Intended audience

This document is intended for technology officers, enterprise storage managers, and storage administrators who want to learn about the different functional features and capabilities of IBM FlashSystem A9000R.

Document conventions

These notices are used in this guide to highlight key information.

Note: These notices provide important tips, guidance, or advice.

Important: These notices provide information or advice that might help you avoid inconvenient or difficult situations.

Attention: These notices indicate possible damage to programs, devices, or data. An attention notice appears before the instruction or situation in which damage can occur.

Related information and publications

You can find additional information and publications related to IBM FlashSystem A9000R on the following information sources.

- IBM FlashSystem A9000R on IBM® Knowledge Center (ibm.com/support/knowledgecenter/STJKN5) – on which you can find the following related publications:
  - IBM FlashSystem A9000R – Release Notes
  - IBM FlashSystem A9000R – Deployment Guide
  - IBM FlashSystem A9000R – Command-Line Interface (CLI) Reference Guide
  - IBM FlashSystem A9000 and IBM FlashSystem A9000R – Open API Reference Guide
- IBM Flash Storage and Solutions marketing website (ibm.com/systems/storage/flash)
- IBM Storage Redbooks® website (redbooks.ibm.com/portals/storage)
- IBM Hyper-Scale Manager on IBM Knowledge Center (ibm.com/support/knowledgecenter/SSUMNQ)

IBM Publications Center

The IBM Publications Center is a worldwide central repository for IBM product publications and marketing material.

The IBM Publications Center website (ibm.com/shop/publications/order) offers customized search functions to help you find the publications that you need. You can view or download publications at no charge.

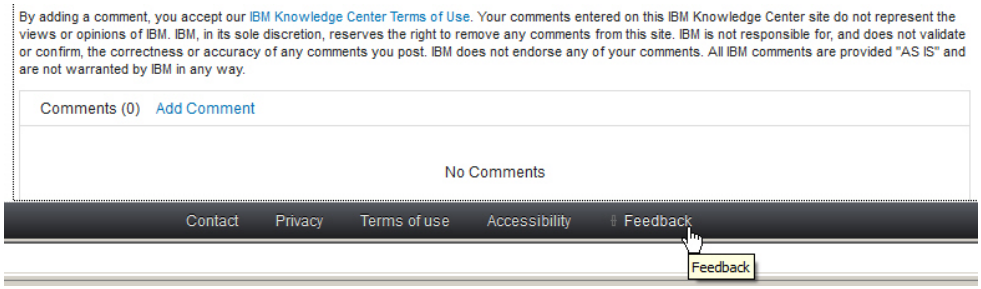
Sending or posting your comments

Your feedback is important in helping to provide the most accurate and highest quality information.

Procedure

To submit any comments about this guide:

- Go to IBM FlashSystem A9000R on IBM Knowledge Center (ibm.com/support/knowledgecenter/STJKN5), drill down to the relevant page, and then click the **Feedback** link that is located at the bottom of the page.



  The feedback form is displayed and you can use it to enter and submit your comments privately.
- You can post a public comment on the Knowledge Center page that you are viewing, by clicking **Add Comment**. For this option, you must first log in to IBM Knowledge Center with your IBMid.
- You can send your comments by email to starpubs@us.ibm.com. Be sure to include the following information:
  - Exact publication title and product version
  - Publication form number (for example: SC01-0001-01)
  - Page, table, or illustration numbers that you are commenting on
  - A detailed description of any information that should be changed

  Note: When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Getting information, help, and service

If you need help, service, technical assistance, or want more information about IBM products, you can find various sources to assist you. You can view the following websites to get information about IBM products and services and to find the latest technical information and support.

- IBM website (ibm.com®)
- IBM Support Portal website (ibm.com/storage/support)
- IBM Directory of Worldwide Contacts website (ibm.com/planetwide)
- IBM developerWorks Answers website (www.developer.ibm.com/answers)
- IBM service requests and PMRs (ibm.com/support/servicerequest/Home.action)

Use the Directory of Worldwide Contacts to find the appropriate phone number for initiating voice call support. Voice calls arrive to Level 1 or Front Line Support.

Chapter 1. Introduction

IBM FlashSystem A9000R is a high-end, grid scale, all-flash storage platform that delivers ultra-fast storage together with mission-critical features, including inline deduplication and data reduction by compression, smart scaling, distributed data, automatic load balancing, and a multitude of advanced enterprise-class features and capabilities.

Figure 1. IBM FlashSystem A9000R unit

As a pre-integrated rack-based offering, IBM FlashSystem A9000R aggregates grid elements (each containing two grid controllers and one flash enclosure) within a 42U integrated rack solution. Grid elements are interconnected by integrated InfiniBand switches, forming a scale-out grid fabric that delivers consistent, predictable high performance and ultra-low latency, even under heavy workloads with full data reduction enabled. The grid architecture enables the system to maintain this performance autonomously by evenly distributing every workload's data across all the system resources in real time. QoS features also help ensure that tenant service levels are not compromised.

In its core architecture, IBM FlashSystem A9000R utilizes IBM FlashCore® technology together with IBM MicroLatency® modules, providing high density, low latency, and high IOPS performance. In addition, IBM FlashSystem Enhanced Endurance Technology reduces flash disk wear-out and ensures long-term durability of the flash storage components, even under heavy workloads.

On top of that, the built-in Spectrum Accelerate™ software provides redirect-on-write space-efficient snapshots, along with asynchronous and synchronous replication to enable granular data protection without increasing costs. To deliver high availability levels that exceed the IBM FlashSystem A9000R 99.999 availability, the system embeds a native implementation of IBM HyperSwap® capability, delivering active-active data access and transparent failover per volume, across IBM FlashSystem A9000R and IBM FlashSystem A9000 arrays and across data center.

Built as a rack-based grid scale system, IBM FlashSystem A9000R is able to meet your diverse workload requirements with a choice of performance and capacity optimized configurations. The IBM Hyper-Scale Manager allows orchestration of private and hybrid multi-tenant cloud environments at very large scales. The IBM Hyper-Scale Manager provides the ability to manage multiple IBM FlashSystem A9000R, IBM FlashSystem A9000, IBM XIV® and IBM Spectrum Accelerate™ solutions from a single pane of glass.

IBM FlashSystem A9000R is ideal for large enterprises that rely on fast, redundant, and high-capacity data storage, offering high service levels for dynamic workloads and easy hyper-scaling, while supporting multi-tenant environments, flexible consumption models, and robust cloud automation and integration capabilities. IBM FlashSystem A9000R offers security and data protection through advanced remote mirroring, hot encryption, and self-healing mechanisms.

Cloud economics and agility are adopted with ready A9000R solutions for Kubernetes and iSwarm container environments, as well as IBM Cloud Private, VMware, OpenStack, and Microsoft environments, at no additional cost.

Architecture

The high-end architecture of IBM FlashSystem A9000R comprises both the hardware architecture and logical architecture.

The primary system components, also referred to as grid elements, include:

- **Flash enclosures** – The flash hardware units upon which the data is written. FlashSystem A9000R includes a minimum of 2 and up to 6 flash enclosures in a single rack.
- **Grid controllers** – The compute servers through which all data is processed. The grid controllers also provide the data reduction function, cache function, and

host interfaces. FlashSystem A9000R includes two grid controllers for every single flash enclosure, so the total number of grid controllers is:

Table 1. Total number of grid controllers

Model 415	Models 425 and U25
4, 6, 8, 10, or 12.	4, 6, or 8.

- **Back-end interconnect** – The InfiniBand infrastructure that connects between all flash enclosures and the grid controllers. FlashSystem A9000R includes dedicated InfiniBand switches for ultra-fast interconnect between all system components.

The following figure shows how the flash enclosure units are coupled and stacked in different configurations of models 415, 425, and U25:

Figure 2. IBM FlashSystem A9000R main components

Note: Data is automatically processed and distributed through all grid elements that are included in a FlashSystem A9000R system, and is not limited to any particular set of two grid controllers and one flash enclosure. Starting from software version 12.0.3, IBM FlashSystem A9000R supports non-disruptive addition of sets of grid element sets that include two grid controllers and one flash enclosure, up to the full physical capacity of the rack system.

Flash enclosures

Each flash enclosure comprises two fully redundant flash canisters, 12 MicroLatency flash modules, two battery modules, and two power supply units.

Each of the two flash canisters includes a RAID 5 controller, interface controller, management ports, and fans. The two redundant canisters are hot-swappable.

The usable capacity of MicroLatency flash modules is shown in the following table:

Table 2. Usable capacity of MicroLatency flash modules

Model 415	Models 425 and U25
12 hot-swap 1.2, 2.9, or 5.7 TB IBM MicroLatency modules	12 hot-swap 3.6, 8.5, or 18 TB IBM MicroLatency modules

The maximum usable RAID 5 array capacity is 180 TB.

The flash enclosures of model 425 feature 3D TLC (Triple Level Cell) NAND MicroLatency modules, enabling dramatic improvement in density and TCO (total cost of ownership).

IBM FlashCore technology

The flash enclosures deliver the full range of the unique IBM FlashCore technology, as summarized in the following figure.

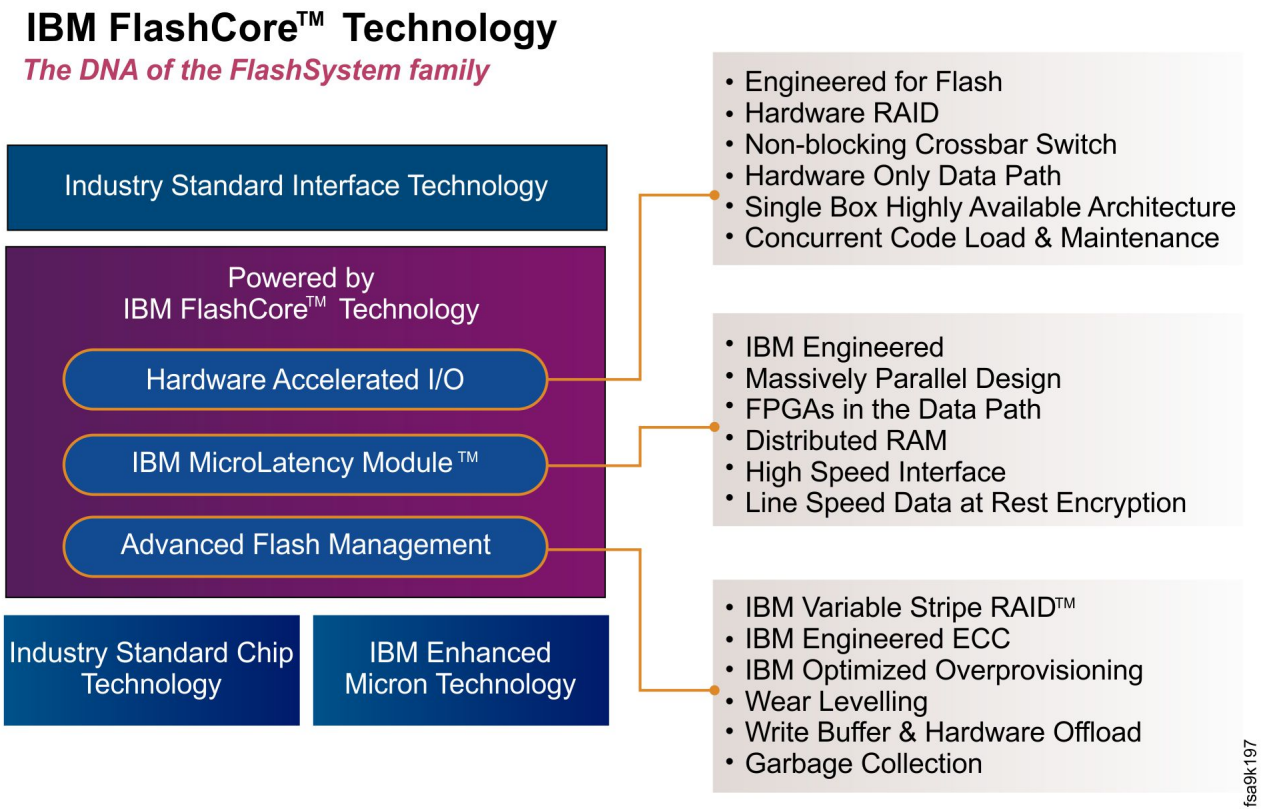


Figure 3. IBM FlashCore technology benefits

Grid controllers

Each grid controller is a high-end Intel Xeon based compute server, which acts as a core component that provides the interface and advanced compute functions.

The grid controllers also provide cache to accelerate both read and write operations, and are responsible for carrying out the entire data reduction operation (see Chapter 2, “Flash-optimized data reduction,” on page 9). Data compression is provided by dedicated hardware accelerator cards.

Each grid controller is designed for high reliability and for modularity, so that components can be replaced in case of a failure, without interrupting the operation of the controller. By design, the grid controller is an isolated failure domain. Any failure or a maintenance action that requires shutdown of a grid controller, does not affect either the overall system operation or the protection status.

IBM FlashSystem A9000R storage systems with enhanced grid controllers feature FC-NVMe ready adapters (see the Fibre Channel NVMe connectivity hardware announcement). NVM Express® (NVMe) allows servers to leverage the native parallelism of today's SSD offerings, reduces overall I/O overhead, and increases bandwidth. FC-NVMe enables NVMe over a Fibre Channel (FC) network fabric, thus combining the benefits of all-flash SAN storage with NVMe performance over existing infrastructure. A system is FC-NVMe ready if it requires a future software update to provide full FC-NVMe support.

To summarize, each grid controller contains:

- 56 Gbps InfiniBand HCAs
- Host interface ports
- Two hot-swappable hard disk drives (HDDs) that contain the system microcode and store various system logs and events
- Two hot-swappable solid-state drives (SSDs) that are used as vault devices
- Two hot-swappable battery backup units (BBUs)
- Two hot-swappable power supplies
- Field-replaceable fans
- Field-replaceable DIMMs

Back-end interconnect

Internal communication (interconnect) between grid controllers or flash enclosures is carried over a 56 Gbps Fourteen Data Rate (FDR) InfiniBand network infrastructure, with full redundancy.

In FlashSystem A9000R, the grid controllers and the flash enclosures are connected through redundant InfiniBand switches, and redundant cabling. The InfiniBand switches are also linked to each other.

The following figure describes the redundant InfiniBand connectivity with four grid controllers and two flash enclosures as an example, with redundant InfiniBand switches and cabling.

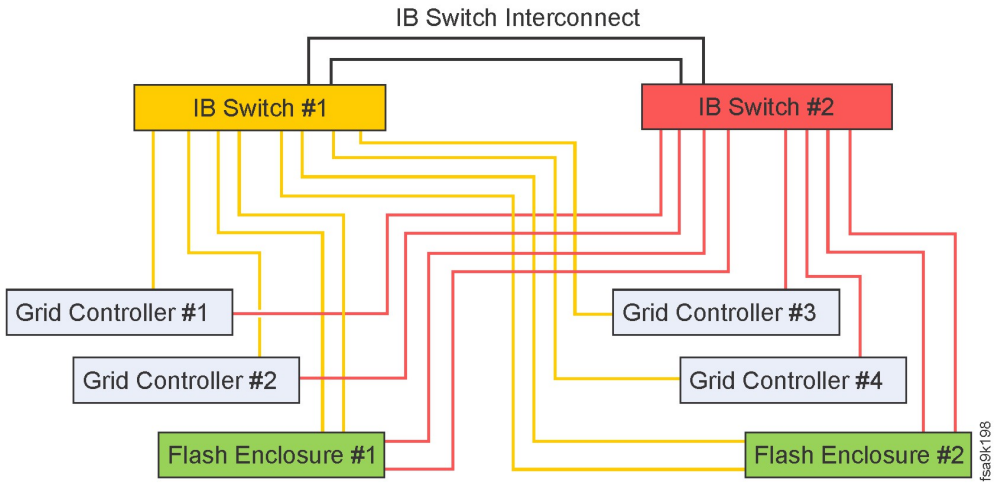


Figure 4. InfiniBand switch connectivity in FlashSystem A9000R

Logical architecture

The IBM FlashSystem A9000R logical architecture is built on the IBM Spectrum Accelerate software, with many added enhancements to optimize the software stack for use with flash storage.

An important feature and differentiation in this new architecture is the software separation between the computation and storage resources of the product. In terms of the physical architecture the compute functions are implemented in grid controllers, while the storage function is implemented in flash enclosures.

The separation of the cache, compute, and interface functions from the storage resource separates load balancing across compute and storage resources. It also enables support for a different resiliency scheme in which cache data is triplicated. This cache data protection is unique to IBM FlashSystem A9000R.

Another significant enhancement to the underlying Spectrum Accelerate software is the data reduction feature, that combines pattern matching and removal, deduplication, and compression. IBM FlashSystem A9000R also offers significant CPU processing power and RAM memory, which allows efficient processing of real-time (inline) deduplication and compression.

Scale-out grid architecture

Scale-out grid architecture allows the increase of the system's capacity without losing performance.

The system supports scale-out features that are to become available in future software (microcode) versions. In such future versions, scale-out would be achieved by interconnecting any single system with other IBM FlashSystem A9000 or A9000R systems, by using IBM Hyper-Scale Manager together with the IBM Hyper-Scale Mobility feature.

Functionality

IBM FlashSystem A9000R delivers a range of advanced storage functional features.

These functional features include:

- Data reduction: pattern removal, deduplication, and compression
- Space-efficient snapshots and consistency groups
- Redirect-on-Write (ROW)
- Host rate limiting: Quality of Service (QoS) performance classes
- Synchronous remote mirroring
- Cross-generation asynchronous replication between A9000R systems and XIV Gen3 systems
- Multi-tenancy
- HyperSwap
- Data-at-rest encryption
- Remote support and proactive support
- Advanced statistics, including:
  - Capacity statistics
  - System performance metrics, including history metrics of no less than 300 days

The following figure illustrates how the first three functional features (see list above) are working in concert to provide top-of-the-line capabilities in an all-flash storage system.

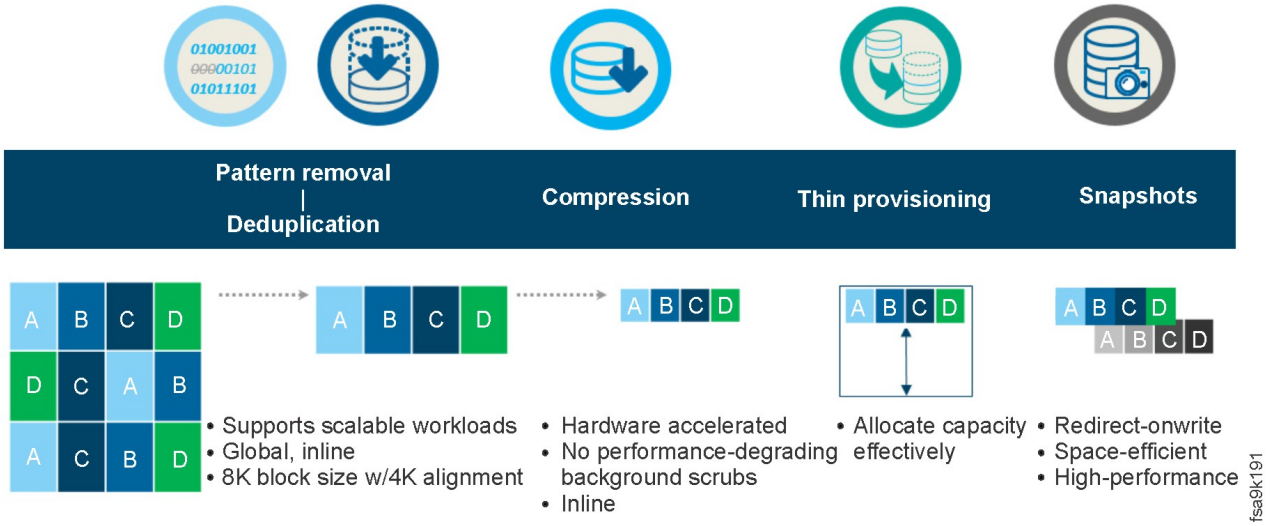


Figure 5. Data reduction and protection

Specifications

For the IBM FlashSystem A9000R hardware and other specifications, refer to the marketing data sheet or to the IBM FlashSystem A9000R Deployment Guide.

Note: For information about specification-related functional boundaries, such as maximum number of volumes and hosts per storage system, refer to the IBM FlashSystem A9000R specification sheet document, which can be provided to IBM customers under non-disclosure agreement (NDA).

Performance

IBM FlashSystem A9000R provides dynamic performance and ultra-low latency of IBM FlashCore technology, integrated with an innovative grid architecture, extensive scalability, and industry-leading IBM software-defined storage capabilities.

As a high-end all-flash storage platform, IBM FlashSystem A9000R is designed to address the most demanding requirements of enterprise clouds. The system architecture and range of grid components enable the system as a whole to deliver consistent microsecond latency with market-leading performance.

Chapter 2. Flash-optimized data reduction

Ultra-fast and always-on data reduction is a fundamental key feature of the storage system.

The storage system uses industry-leading data reduction technology that combines in-line, real-time pattern matching and removal, deduplication, and compression. Together and in concert, these data reduction mechanisms allow the total physical storage capacity to be a few times larger, without affecting performance even in heavy workloads.

The average data reduction ratio is approximately 5:1. However, the actual ratio could vary depending on the data type, as detailed in the following table.

Table 3. Data reduction ratio for different workloads

Workload type	Data reduction ratio
Virtual desktop infrastructure (VDI)	48:1 (98% savings)
VMware, Linux, and Windows	4:1 (74% savings)
Kernel-based virtual machine (KVM) on Linux	9:1 (89% savings)
Kernel-based virtual machine (KVM) on Windows	2:1 (55% savings)
Database	4:1 (77% savings)

Note: Other types of data environments and workloads may yield different reduction ratios.

Data reduction is implemented below the global cache to ensure very rapid response times, provide a global scope for data reduction services, and allow other data services to be completely unaffected, including snapshots, replication, offload-to-host features, and more.

With constant and automatic data reduction, the storage system's allocation limit is always larger than its raw physical capacity.

The following subsections describe the data reduction process and individual stages.

Data reduction stages

Each grid controller in the system dedicates some CPU processing capacity and memory resources for the purpose of data reduction.

The on-the-fly data reduction stages are:

1. Pattern detection and removal
2. Inline deduplication
3. Inline compression

The following figure illustrates how original data is reduced after each stage, before it is written to the flash disk.

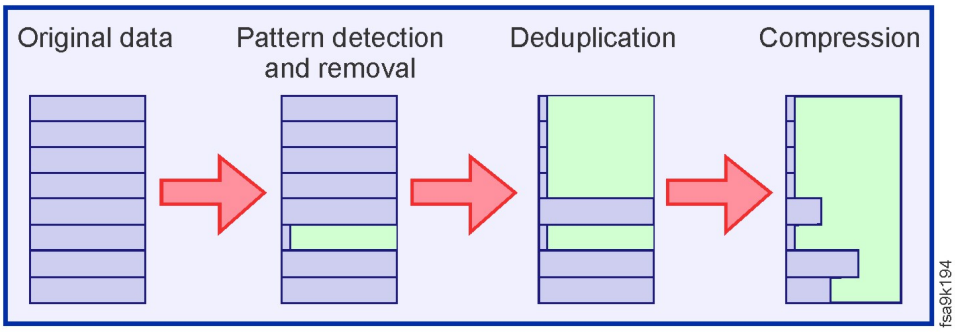


Figure 6. Data reduction stages

The system's data reduction node implements the data deduplication and compression functions.

The three data reduction stages are carried out according to a certain flow logic, as detailed in the following figure.

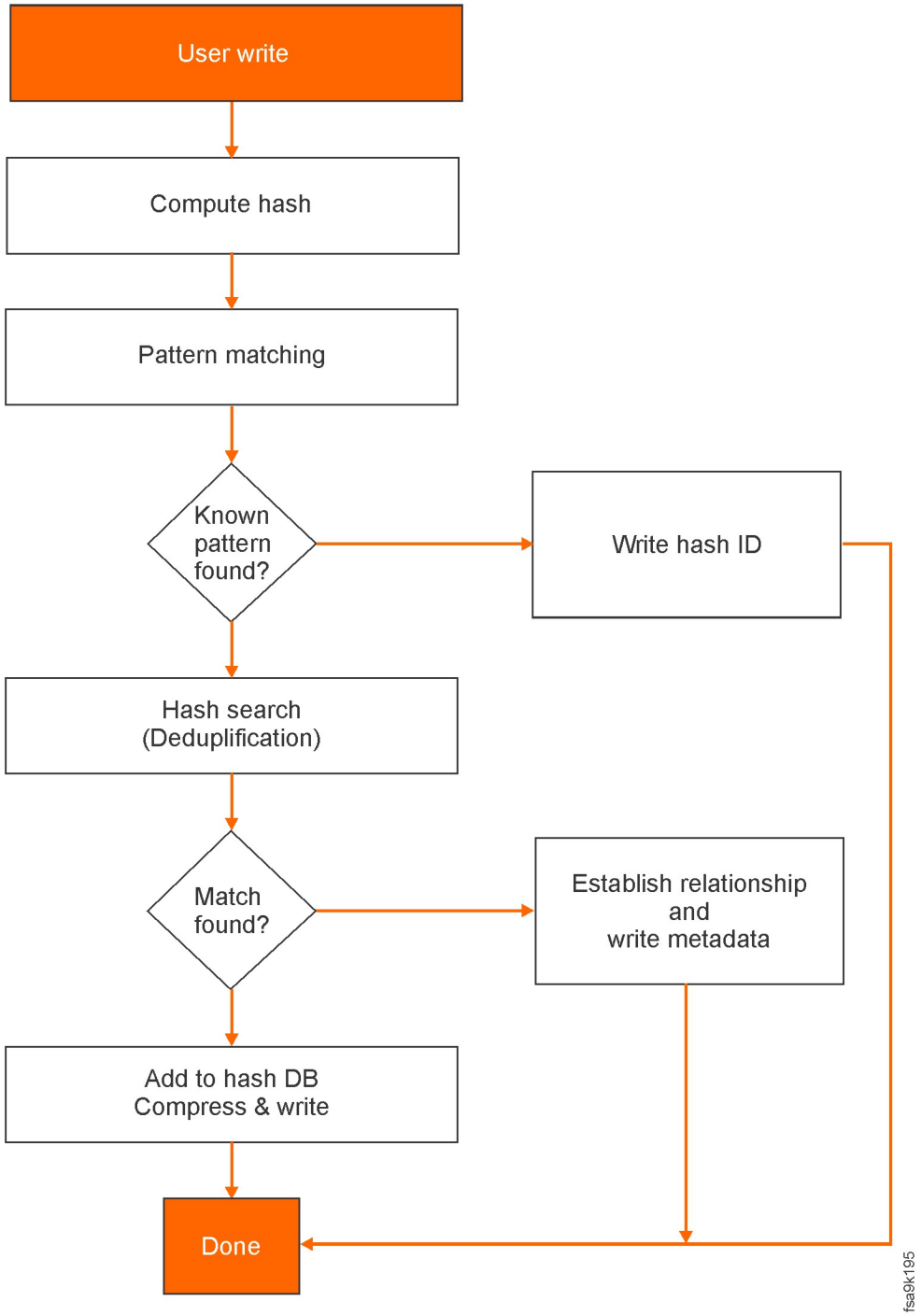


Figure 7. Data reduction process logic

Pattern detection and removal

Pattern detection and removal is the first stage in the data reduction process.

Pattern matching mechanisms match incoming host writes with a preconfigured set of known patterns stored in the system. When a write is processed, it is split into 8 KB blocks, as shown in the following figure.

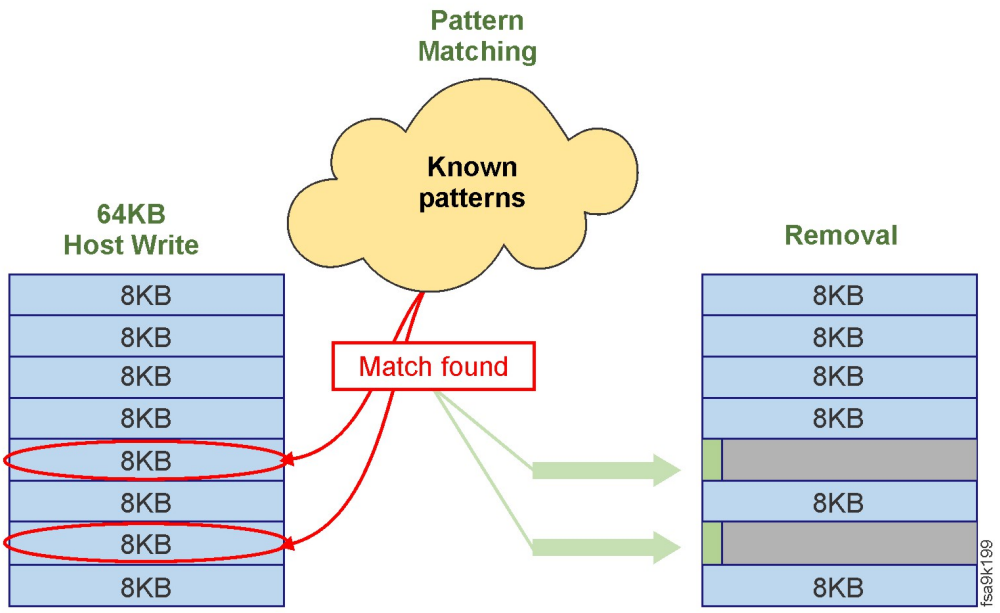


Figure 8. Pattern detection and removal

Each block is then hashed and the hash value, also known as a fingerprint, is compared to a table of well-known hashes. If a match is found, then the corresponding pattern ID, which is only 2 bytes is stored. Any match found at that stage is replaced with internal markings (a hash).

Inline deduplication

The inline deduplication is the second data reduction stage that processes and consolidates data before it is written to disk.

Hashing and hash comparison is performed on-the-fly. The benefit of inline deduplication is that duplicate chunks are never written to the destination disk system.

The following figure illustrates how original data is reduced via the deduplication process by using hashing and metadata.

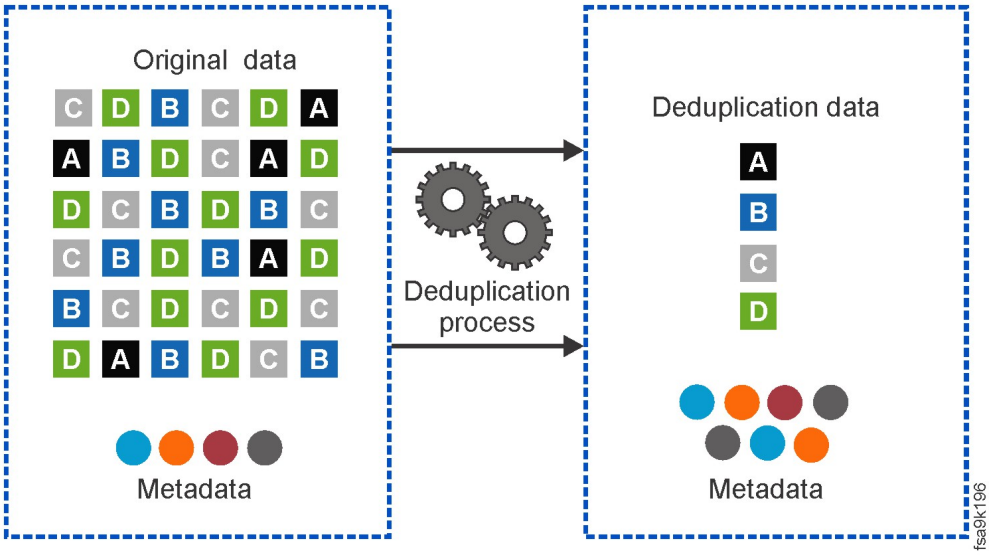


Figure 9. Deduplication process

In this process:

- Hashes are calculated only once (upon write)
- Hashes are stored in metadata
- Data is compared by comparing the hash
- The chunk size is 8KB, although 4KB alignment is also supported and allows for a higher deduplication ratio

Data reduction with deduplication alone could reach to up to 30:1 (96.7% savings).

Inline compression

Inline compression is the last data reduction stage.

The IBM patented compression technology used by the storage system is based on a data compression algorithm that operates in real time.

The major difference between traditional compression and IBM compression technique is in the size of data blocks that are written to the storage device. The IBM compression technique uses fixed-size writes. This method enables an efficient and consistent method to index the compressed data, because it is stored in fixed-size containers.

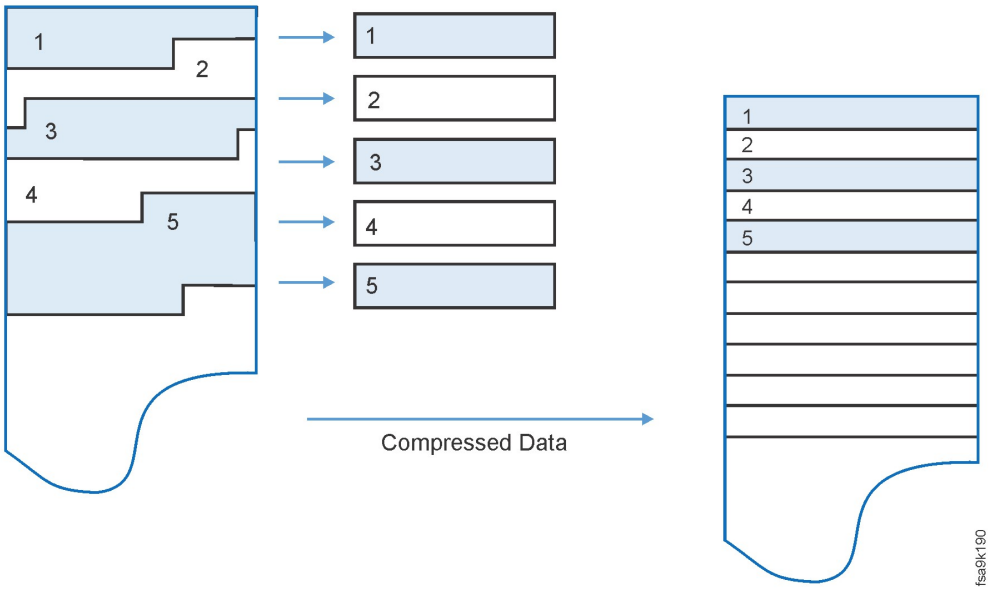


Figure 10. IBM compression method - using fixed-size writes

The compression ratio of a data block depends on how many repetitions can be detected within the block. The number of repetitions is affected by how many bytes present in the block are related to each other. The relation between bytes is driven by the format of the compressed object.

Chapter 3. Flash-optimized data protection

The constant operation and availability of the system relies on the ratio of its functionality time to the total time it is required to function.

The system's high availability (HA) features pertain to both the hardware and software components. Hardware components are hot-swappable and support instant failover capabilities, allowing the system to have no single point of failure (SPOF).

In addition to 2D Flash RAID protection and Variable Stripe RAID data protection (see "Two-dimensional flash RAID"), the system's flash enclosure incorporates additional reliability features:

- Error correction codes to provide bit-level reconstruction of data from flash chips (see "Scrubbing mechanism" on page 17).
- Checksum and data integrity fields designed to protect all internal data transfers within the system.
- Overprovisioning to enhance write endurance and decrease write amplification.
- Wear-leveling algorithms balance the number of writes among flash chips throughout the system.
- Sweeper algorithms help ensure that all data within the system is read periodically to avoid data fade issues.

Additional system protection mechanisms include:

- Advanced vaulting mechanism, including live vaulting (see "Vaulting mechanism" on page 17).
- All cached data is protected using triplication (3 copies).
- Software nodes are redundant with auto-restart of hanging nodes.
- Any unresponsive data node is expelled from the cluster automatically.
- Upon any performance degradation, events are issued and reported.
- System temperature monitoring with warning events generation.
- Power monitoring, including battery unit conditioning (calibration) for increased capacity and lifespan.

Two-dimensional flash RAID

The combination of IBM Variable Stripe RAID and system-level RAID 5 protection across IBM MicroLatency modules (within a flash enclosure) is called two-dimensional (2D) flash RAID.

Two-dimensional (2D) flash RAID consists of IBM Variable Stripe RAID and system-wide RAID 5.

Variable Stripe RAID technology helps reduce downtime and maintain performance and capacity in the event of partial or full flash chip failures. Failures of large capacity granularity, such as an entire flash chip or a plane or a die of a flash chip, are isolated, reduced to the absolute minimum affected area, then bypassed.

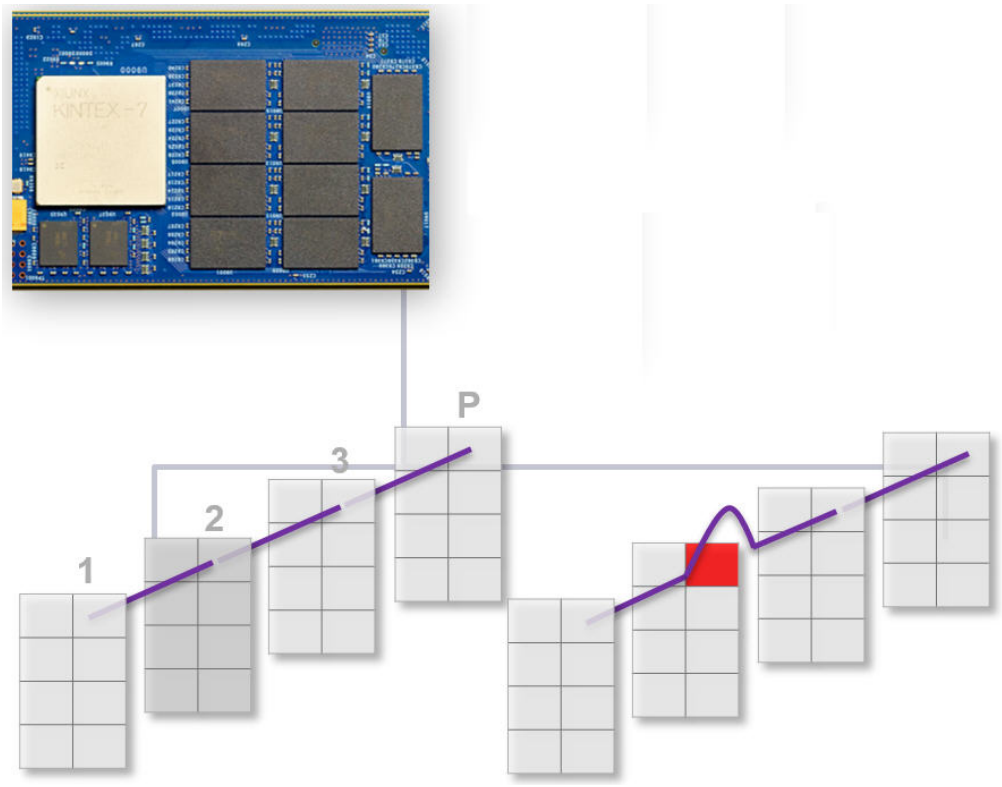


Figure 11. Variable Stripe RAID technology

System-wide RAID 5, with easily accessed hot swappable flash modules, helps promote availability. RAID 5 configurations provide a high degree of redundancy with Variable Stripe RAID and RAID 5 protection. RAID 5 data protection includes one IBM MicroLatency module dedicated as parity and another as a dedicated hot spare. The maximum capacity utilization for RAID 5 is provided by using 12 IBM MicroLatency modules.

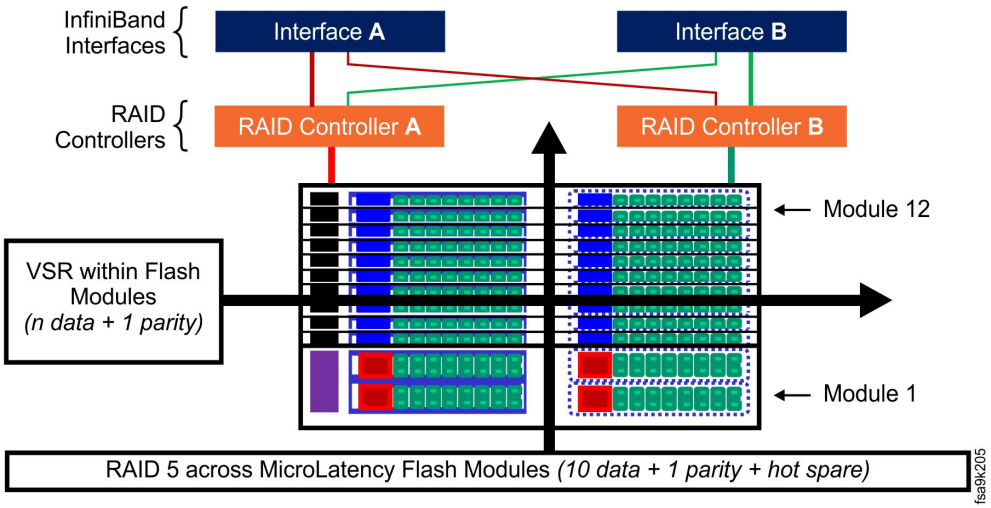


Figure 12. 2D RAID protection mechanism

The system's flash enclosure level RAID 5 complements the Variable Stripe RAID technology implemented within each MicroLatency module, providing protection

against data unavailability resulting from MicroLatency module failures. It also allows data to be rebuilt onto a hot spare flash module, so that any individual MicroLatency module could be replaced without data disruption.

Vaulting mechanism

Each grid controller contains two 400 GB enterprise-grade SSD disks for saving cache data in case of power loss, in a process called *vaulting*.

In addition, the SSD disks are used to save metadata and system configuration information on a continuous basis, in a process called *live vaulting*. During a normal shutdown procedure, the system's microcode also writes to the SSD disks all configuration data, metadata, and cache data that has not yet been stored in the permanent flash enclosure storage.

As shown in the following figure, the cache data is mirrored three times across different grid controllers, without requiring mirroring within the grid controller.

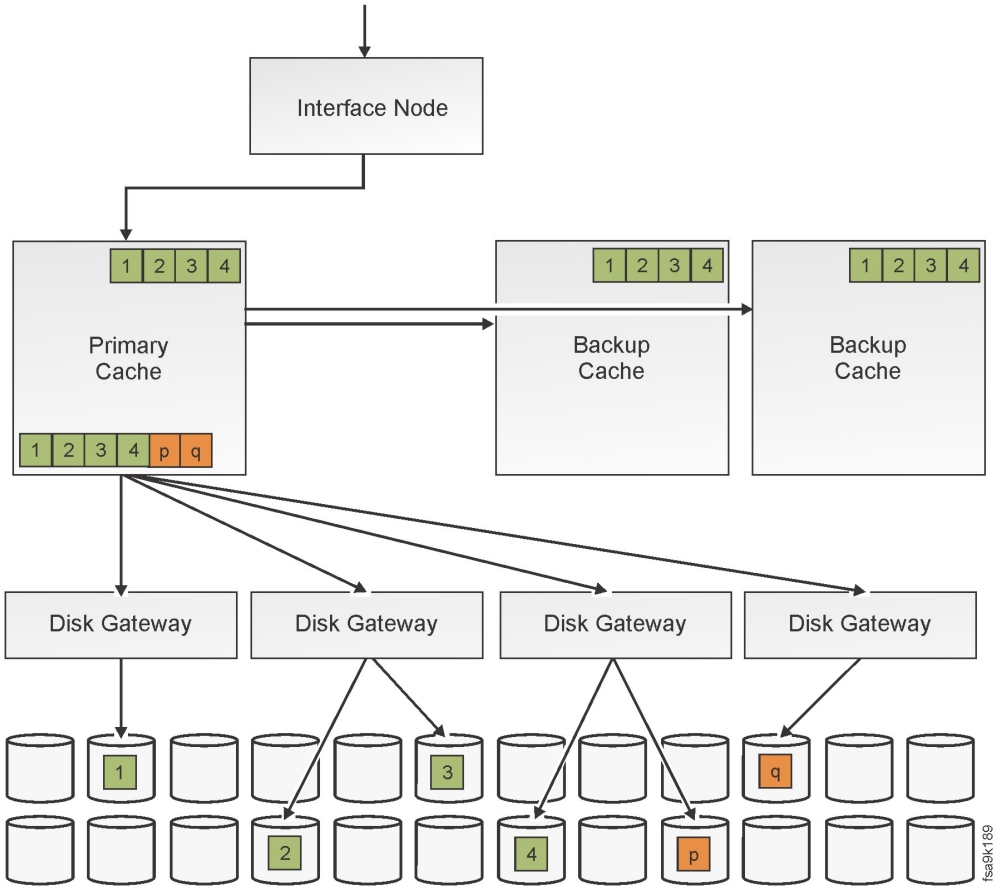


Figure 13. 3 copies of cache data

Scrubbing mechanism

Scrubbing is a background process that systematically verifies there are no data inconsistencies, and, if possible, amends the detected errors.

Physical data scrubbing is performed autonomously and internally by the flash enclosure. The scrubbing process verifies the consistency of the RAID 5. Error correction is performed for recoverable errors (single error), and no correction is performed for unrecoverable errors (double error).

For unrecoverable errors, an event is generated and reported, but the flash enclosure remains up and running.

Physical scrubbing is complemented by virtual scrubbing which verifies the data through RACE. This additional scrubbing process can also make an effort to correct the detected errors (for example, by using mirroring). Virtual scrubbing, in its turn, is combined with remote scrubbing, which also sequentially reads the data through RACE. When the volume is primary in remote mirroring (see Chapter 11, "Synchronous remote mirroring," on page 55), remote scrubbing compares the content on the partition to its counterpart on the remote target. If the partition contents are not identical, an event is issued.

Chapter 4. Flash-optimized data path

The hardware-only data path design of the flash enclosure eliminates latency at the system's software layer. The system's data path is completely independent of the control path.

Data traverses through the flash enclosure through field-programmable gate arrays (FPGAs), preventing wasted cycles on interface translation, protocol control, or tiering. FPGAs are included in the interface cards, RAID controllers, and MicroLatency modules.

In addition, all data passes through three nodes:

- Interface nodes
- Cache nodes
- Data reduction nodes

These nodes reside on each grid controller. The system load is uniformly distributed between all the grid controllers, resulting with a statistically balanced load. All data nodes communicate with each other, so that data entering the system through a certain interface node can be directed to any of the cache nodes, in any of the grid controllers.

The data flow includes the following stages:

1. I/O requests enter the system as SCSI commands on any of the interface nodes running on the grid controllers. The I/O requests are forwarded to the cache nodes based on a distribution table, which maintains rules to ensure that the I/O and capacity loads are equally distributed.
2. The cache data is copied three times across different grid controllers, providing protection in case of any unexpected failure.
3. The cache node directs the data to any of the data reduction nodes on any of the grid controllers.
4. The data reduction node performs data reduction and then moves it back to one of the cache nodes.
5. The cache node then acts as a gateway, performing a read/write operation to the flash enclosure.

The following figure illustrates these stages:

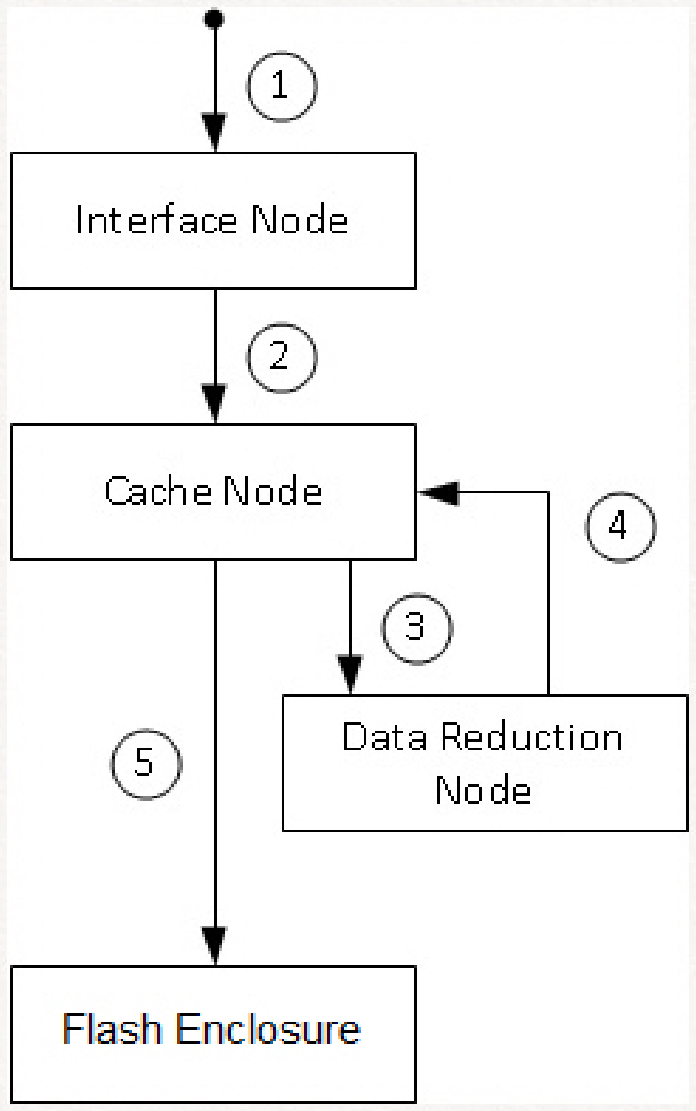


Figure 14. Data path through the different nodes

Upon any failure in one of the grid controllers, or upon detection of a slow node, the system automatically and seamlessly re-adjusts the load distribution, and the data of the failed/slow component is shared among the rest of the nodes, to avoid performance degradation and to re-balance the system.

FlashSystem A9000 systems include only one flash enclosure. However, on FlashSystem A9000R systems that include more than one flash enclosure, data can be moved from any grid controller to any flash enclosure. This mesh architecture enables the system to evenly distribute workloads in the system.

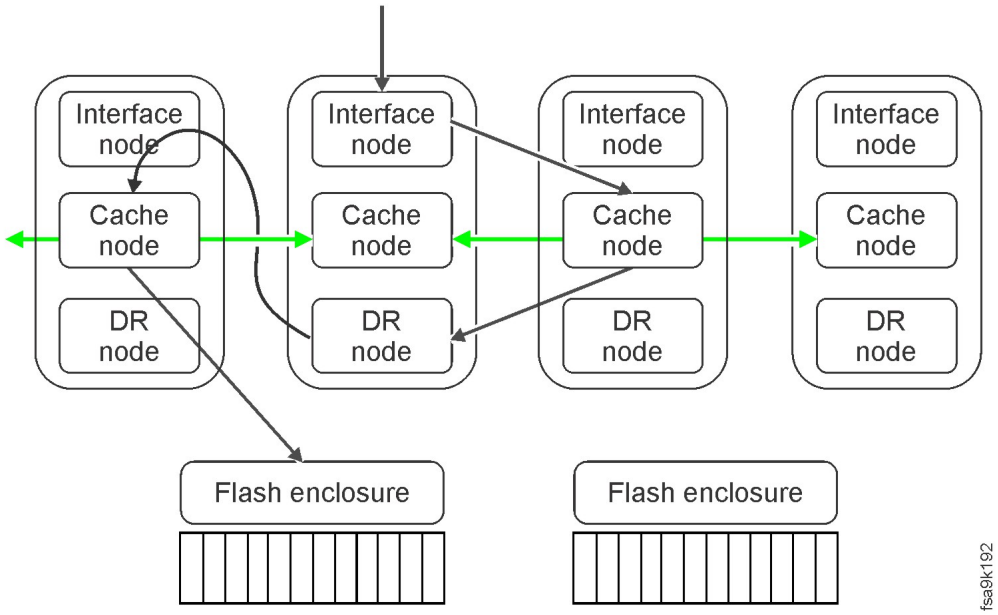


Figure 15. Data path mesh architecture

Chapter 5. Capacity management

Managing and utilizing the storage system's capacity is a key aspect of the system management.

Capacity management hierarchy in a storage system is shown in the diagram below:

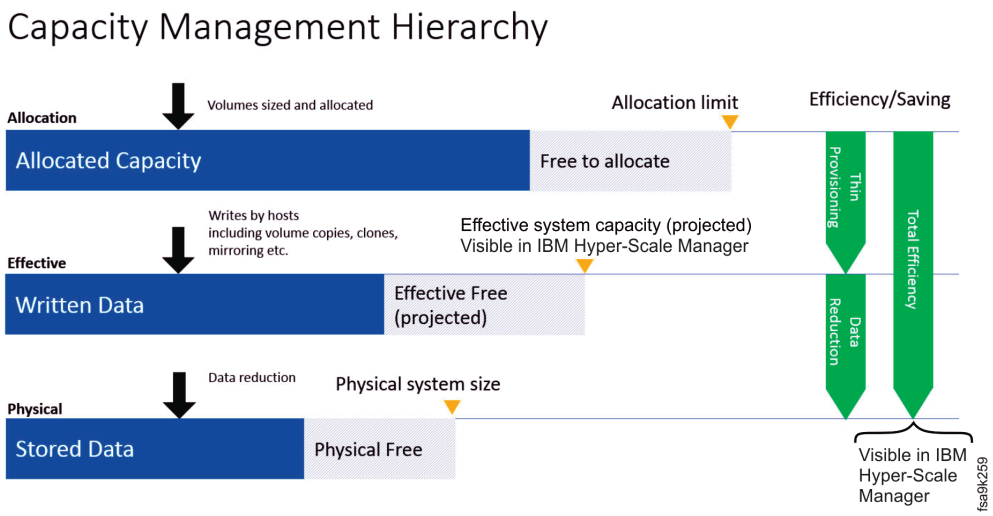


Figure 16. Capacity management hierarchy

- **Physical size** – Represents the physical amount of storage space that is available in the system. In systems with more than a single flash enclosure, the physical capacity is the sum of capacity of all flash enclosures combined.
- **Effective capacity** – Represents the total data written by hosts to volumes and snapshots, before applying data reduction (see Chapter 2, "Flash-optimized data reduction," on page 9). This includes volume copies, clones, mirroring, and so on.
- **Allocation limit** – Represents the actual available-for-use capacity after data reduction. The allocation limit varies depending on the type of data being written, and the data reduction ratio that is achieved for that type of data. With constant and automatic data reduction, the allocation limit is always much larger than the system's raw physical capacity. The segment of the allocation limit that is currently allocated to volumes and snapshots is referred to as **allocated capacity** (see also "Allocation limit" on page 24).

Capacity management includes the following aspects:

- Provisioning – Defining resources and assigning these resources to storage objects (domains, storage pools, volumes, snapshots).
- Accounting – Tracking resource consumption at various levels, and making sure that provisioned boundaries are observed.
- Statistics – Collecting capacity usage statistics over different periods of time.
- Monitoring – Defining capacity usage thresholds and alerts.

When managing the system capacity fromIBM Hyper-Scale Manager, additional projection and planning capabilities and management options are available. For details, refer to IBM Hyper-Scale Manager on IBM Knowledge Center (ibm.com/support/knowledgecenter/SSUMNQ).

Note: Customers looking for pay-per-use capacity licensing of FlashSystem A9000 or A9000R are advised to consider FlashSystem A9000 or A9000R model U25, a Storage Utility Offering that is based on the same hardware as model 425. For details, refer to the IBM Storage Utility Offering home page (ibm.com/us-en/marketplace/storage-utility-offering).

Thin provisioning

Thin provisioning, or over-provisioning, provides the ability to define logical volume or storage pool sizes that are much larger than the physical capacity of the system.

Capacity associated with specific applications or users can be dynamically increased or decreased per demand, without necessitating an accurate prediction of future needs. Physical capacity is only committed when the associated applications execute writes, and not when the logical volume is initially allocated. In other words, thin provisioning is the virtualization of the capacity from the underlying hardware allocation.

Because the total system capacity is designed as a globally available pool, thin provisioned resources share the "buffer" of free space. This approach results in highly efficient aggregate capacity use without pockets of inaccessible unused space.

However, because the storage system already implements data reduction that is carried out automatically (see Chapter 2, "Flash-optimized data reduction," on page 9), thin provisioning is provided by default, because the allocation limit is already at least 5 times greater than the physical capacity of the system.

The allocatable capacity in the system can be administratively portioned into separate and independent storage pools (see Chapter 7, "Storage pools," on page 41). The storage administrator can create volumes (along with the reserved snapshot space) whose total capacity is less than or equal to the size of the pool. At the same time, the data reduction methods of deduplication and compression are reducing the amount of data actually written to the flash enclosure.

Allocation limit

Increase in the physical capacity of a single storage system or a group of scaled-out storage systems results in an increase of the maximum allocation limit.

Physical capacity addition is followed by a process of redistribution (restriping) of the existing data across the bigger physical capacity. Only at the end of this phase-in process will the physical capacity be updated to reflect the expanded physical and effective capacity.

The allocation limits for different configurations of IBM FlashSystem A9000R are listed in the table below.

Table 4. Allocation limits for various IBM FlashSystem A9000R configurations

Model	Number of grid controllers	Allocation limit	Number of grid elements
415	4	1400 TB	2
	6	2000 TB	3
	8	2600 TB	4
	10 and more	3000 TB	5 and more
425 and U25	4	2400 TB	2
	6	3600 TB	3
	8	4800 TB	4

Chapter 6. Volumes and snapshots

Volumes are the basic storage data units in the storage system. Snapshots of volumes can be created, where a snapshot of a volume represents the data on that volume at a specific point in time.

Volumes can also be grouped into larger sets called consistency groups (see Chapter 8, "Consistency groups," on page 43) and storage pools (see Chapter 7, "Storage pools," on page 41).

The basic hierarchy is as follows:

- A volume can have multiple snapshots.
- A volume can be part of one and only one consistency group.
- A volume is always a part of one and only one storage pool.
- All volumes in a consistency group must belong to the same storage pool.

Volume function and lifecycle

Volume is the basic data container that is presented to the hosts as a logical disk.

The term *volume* is sometimes used for an entity that is either a volume or a snapshot. Hosts view volumes and snapshots through the same protocol. Whenever required, the term *Primary volume* is used for a volume to clearly distinguish volumes from snapshots.

Each volume has two configuration attributes: a name and a size. The volume name is an alphanumeric string that is internal to the storage system and is used to identify the volume to both the GUI and CLI commands. The volume name is not related to the SCSI protocol. The volume size represents the number of blocks in the volume that the using host detects.

A volume is defined within the context of only one storage pool (see Chapter 7, "Storage pools," on page 41). Because storage pools are logical constructs, a volume and any snapshots associated with it can be moved to any other storage pool (within the same domain) if there is sufficient space within the target storage pool.

Volumes in FlashSystem A9000 and A9000R

As a benefit of the system virtualization, there are no limitations on the associations between logical volumes and storage pools. Moreover, manipulation of storage pools consists exclusively of metadata transactions and does not trigger any copying of data. Therefore, changes are completed instantly and without any system performance degradation.

The storage system uses the grid concept and distributes volume data evenly across hardware storage resources. Volumes are distributed evenly across all flash enclosures using partitions and each partition is 16 MB in size.

The system also uses the concept of allocation unit (AU) size for volumes, which is set at 103 GB. Minimum volume size that can be created is 1 GB. However, volumes that are created with a specified size about five percent or less, smaller

than the AU size or multiples of allocation unit size will be rounded to multiples of AU size. For example, creating a volume, specifying a 98 GB size creates a volume of 103 GB on the system.

Volume management and lifecycle

The following management options are available for volumes:

Create Defines the volume using the attributes you specify

Resize Changes the virtual capacity of the volume.

Copy Copies the volume to an existing volume or to a new volume.

Format
Clears the volume.

Lock Prevents hosts from writing to the volume.

Unlock
Allows hosts to write to the volume.

Rename
Changes the name of the volume, while maintaining all of the volumes previously defined attributes

Delete Deletes the volume.

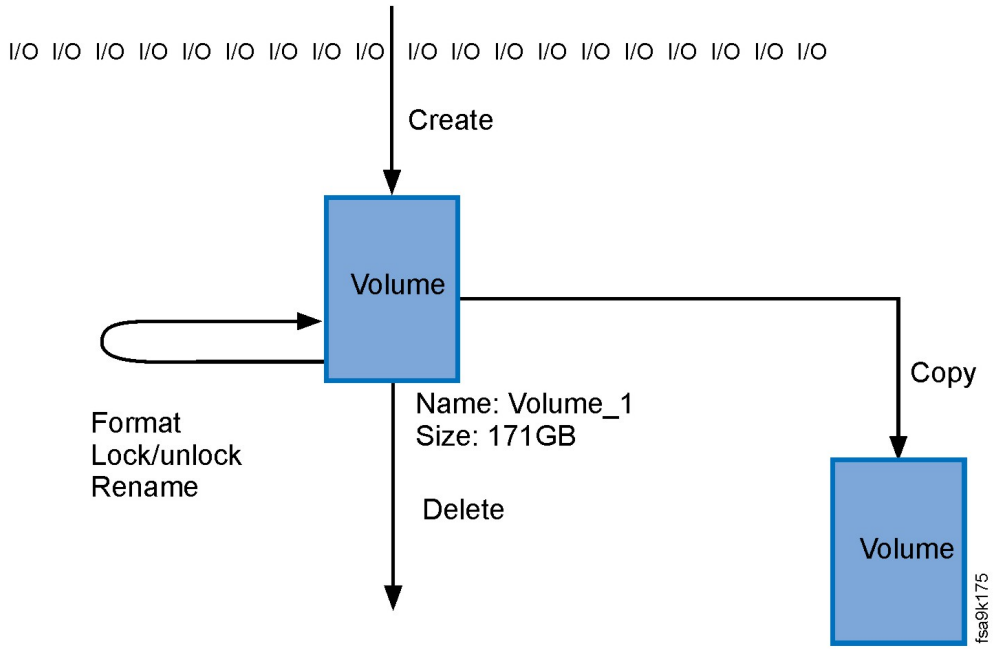


Figure 17. Volume operations

The following query options are available:

Listing volumes
List the details of all volumes, or a specific volume according to a given volume or pool.

Finding a volume based on a SCSI serial number
Display the volume name according to its SCSI serial number.

Snapshot function and lifecycle

A *snapshot* is a logical volume reflecting the contents of a given source volume at a specific point-in-time.

The storage system uses advanced snapshot mechanisms to create a virtually unlimited number of volume copies without impacting performance. Snapshot taking and management are based on a mechanism of internal pointers that allow the Primary volume and its snapshots to use a single copy of data for all portions that have not been modified.

This approach, also known as Redirect-on-Write (ROW) is an improvement of the more common Copy-on-Write (COW), which translates into a reduction of I/O actions, and therefore storage usage. For more information, see "Redirect-on-Write (ROW)" on page 33.

No storage capacity is consumed by the snapshot until the source volume (or the snapshot) is changed.

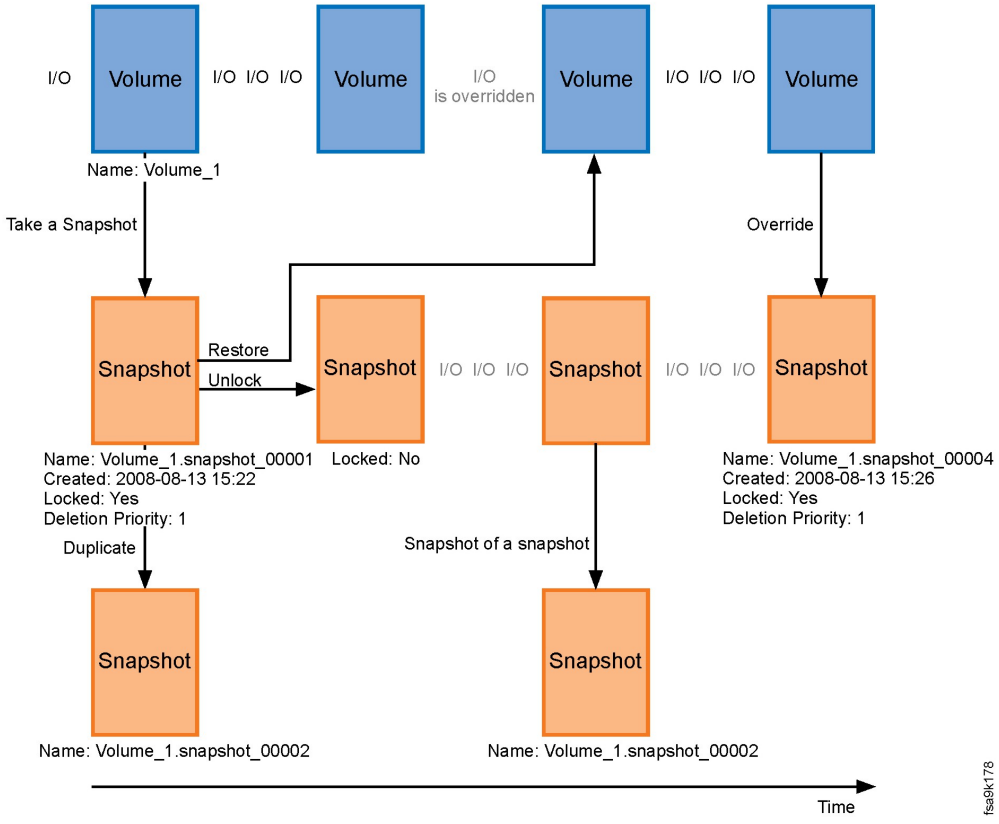


Figure 18. The snapshot life cycle

The following operations are applicable for the snapshot:

Create Creates the snapshot (taking a snapshot).

Restore
Copies the snapshot back onto the volume. The main snapshot functionality is the capability to restore the volume.

Unlocking
Unlocks the snapshot to make it writable and sets the status to Modified. Re-locking the unlocked snapshot disables further writing, but does not change the status from Modified.

Duplicate
Duplicates the snapshot. Similar to the volume, which can be snapshotted infinitely, the snapshot itself can be duplicated.

A snapshot of a snapshot
Creates a backup of a snapshot that was written into. Taking a snapshot of a writable snapshot is similar to taking a snapshot of a volume.

Overwriting a snapshot
Overwrites a specific snapshot with the content of the volume.

Delete Deletes the snapshot.

The following subsections provide additional details about the snapshot lifecycle stages.

Creating a snapshot

First, a snapshot of the volume is taken.

The system creates a pointer to the volume, hence the snapshot is considered to have been immediately created. This is an atomic procedure that is completed in a negligible amount of time. At this point, all data portions that are associated with the volume are also associated with the snapshot.

Later, when a request arrives to read a certain data portion from either the volume or the snapshot, it reads from the same single, physical copy of that data.

When a write I/O request arrives to the Primary volume, a data reference to the overwritten data is created and associated with the Primary volume snapshot. This way, the overwritten data is retained in the snapshot.

Note: For blocks smaller than 8KB, a copy of the overwritten data is created and associated with the Primary volume snapshot instead.

Locking and unlocking a snapshot

Initially, a snapshot is created in a locked state, which prevents it from being changed in any way related to data or size, and only enables the reading of its contents.

This is called an *image* or *image snapshot* and represents an exact replica of the Primary volume when the snapshot was created.

A snapshot can be unlocked after it is created. The first time a snapshot is unlocked, the system initiates an irreversible procedure that puts the snapshot in a state where it acts like a regular volume with respect to all changing operations. Specifically, it allows write requests to the snapshot. This state is immediately set by the system and brands the snapshot with a permanent modified status, even if no modifications were performed. A *modified snapshot* is no longer an image snapshot.

An unlocked snapshot is recognized by the hosts as any other writable volume. It is possible to change the content of unlocked snapshots, however, physical storage space is consumed only for the changes. It is also possible to resize an unlocked snapshot.

Primary volumes can also be locked and unlocked. A locked Primary volume cannot accept write commands from hosts. The size of locked volumes cannot be modified.

Duplicating a snapshot

Authorized users can create a new snapshot by duplicating an existing snapshot.

The snapshot duplicate is identical to the source snapshot. The new snapshot is associated with the Primary volume of the existing snapshot, and appears as if it were taken at the exact moment the source snapshot was taken. For image snapshots that have never been unlocked, the duplicate is given the exact same creation date as the original snapshot, rather than the duplication creation date.

With this feature, a user can create two or more identical copies of a snapshot for backup purposes, and perform modification operations on one of them without sacrificing the usage of the snapshot as an untouched backup of the Primary volume, or the ability to restore from the snapshot.

Creating a snapshot of a snapshot

When duplicating a snapshot that was changed using the unlock feature, the generated snapshot is actually a snapshot of a snapshot.

The creation time of the newly created snapshot is when the command was issued, and its content reflects the contents of the source snapshot at the moment of creation.

After its creation, the new snapshot is viewed as another snapshot of the Primary volume.

Formatting a snapshot or a snapshot group

The format operation deletes the content of a snapshot - or a snapshot group - while maintaining its mapping to the host.

The purpose of the formatting is to allow customers to back up their volumes via snapshots, while maintaining the snapshot ID and the LUN ID. More than a single snapshot can be formatted per volume.

Format operation results

The format operation results with the following:

- The formatted snapshot is read-only
- The format operation has no impact on performance
- The formatted snapshot does not consume space
- Reading from the formatted snapshot always returns zeroes
- It can be overridden
- It can be deleted
- Its deletion priority can be changed

Restrictions

No unlock
The formatted snapshot is read-only and cannot be unlocked.

No volume restore
The volume that the formatted snapshot belongs to cannot be restored from it.

No restore from another snapshot
The formatted snapshot cannot be restored from another snapshot.

No duplicating
The formatted snapshot cannot be duplicated.

No re-format
The formatted snapshot cannot be formatted again.

No volume copy
The formatted snapshot cannot serve as a basis for volume copy.

No resize
The formatted snapshot cannot be resized.

Snapshot formatting example

1. Create a snapshot for each LUN you would like to backup to, and mount it to the host.
2. Configure the host to back up this LUN.
3. **Format the snapshot.**
4. Re-snap. The LUN ID, snapshot ID, and host mapping are maintained.

Restrictions in relation to other operations

Snapshots of the following types cannot be formatted:

Internal snapshot
Formatting an internal snapshot hampers the process it is part of.

Part of a sync job
Formatting a snapshot that is part of a sync job renders the sync job meaningless.

Part of a snapshot group
A snapshot that is part of a snapshot group cannot be treated as an individual snapshot.

Snapshot group restrictions
All snapshot format restrictions apply to the snapshot group format operation.

Additional snapshot attributes

Snapshots have the following additional attributes.

Storage utilization

The storage system allocates space for volumes and their snapshots in a way that whenever a snapshot is taken, additional space is actually needed only when the volume is written into.

As long as there is no actual writing into the volume, the snapshot does not need actual space. However, some applications write into the volume whenever a snapshot is taken. This writing into the volume mandates immediate space allocation for this new snapshot. Hence, these applications use space less efficiently than other applications.

Auto-delete priority

Snapshots are associated with an *auto-delete priority* to control the order in which snapshots are automatically deleted.

Taking volume snapshots gradually fills up storage space according to the amount of data that is modified in either the volume or its snapshots. To free up space when the maximum storage capacity is reached, the system can refer to the auto-delete priority to determine the order in which snapshots are deleted. If snapshots have the same priority, the snapshot that was created first is deleted first.

Name and association

A snapshot can either be taken of a source volume, or from a source snapshot.

The name of a snapshot is either automatically assigned by the system at creation time or given as a parameter of the CLI command that creates it. The snapshot's auto-generated name is derived from its volume's name and a serial number. The same applies when using the management GUI.

The following are examples of snapshot names:

```
MASTERVOL.snapshot_XXXXX
NewDB-server2.snapshot_00597
```

Parameter	Description	Example
MASTERVOL	The name of the volume.	NewDB-server2
XXXXX	A five-digit, zero filled snapshot number.	00597

Redirect-on-Write (ROW)

The storage system uses the Redirect-on-Write (ROW) mechanism.

The following items are characteristics of using ROW when a write request is directed to the Primary volume:

1. The data originally associated with the Primary volume remains in place.
2. The new data is written to a different location on the disk.
3. After the write request is completed and acknowledged, the original data is associated with the snapshot and the newly written data is associated with the Primary volume.

In contrast with the traditional Copy-on-Write (COW) method, with redirect-on-write the actual data activity involved in taking the snapshot is drastically reduced. Moreover, if the size of the data involved in the write request is equal to the system's slot size, there is no need to copy any data at all. If the write request is smaller than the system's slot size, there is still much less copying than with the standard approach of Copy-on-Write.

In the following example of the Redirect-on-Write process, The volume is displayed with its data and the pointer to this data.

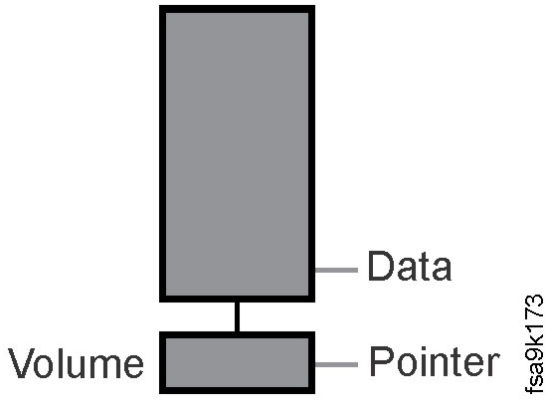


Figure 19. Redirect-on-Write process: the volume's data and pointer

When a snapshot is taken, a new header is written first.

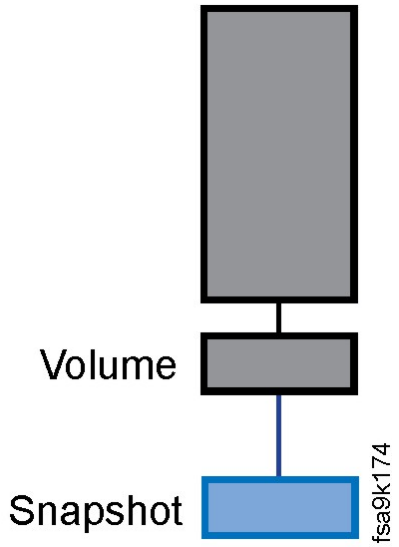


Figure 20. Redirect-on-Write process: when a snapshot is taken, the header is written first

The new data is written anywhere else on the disk, without the need to copy the existing data.

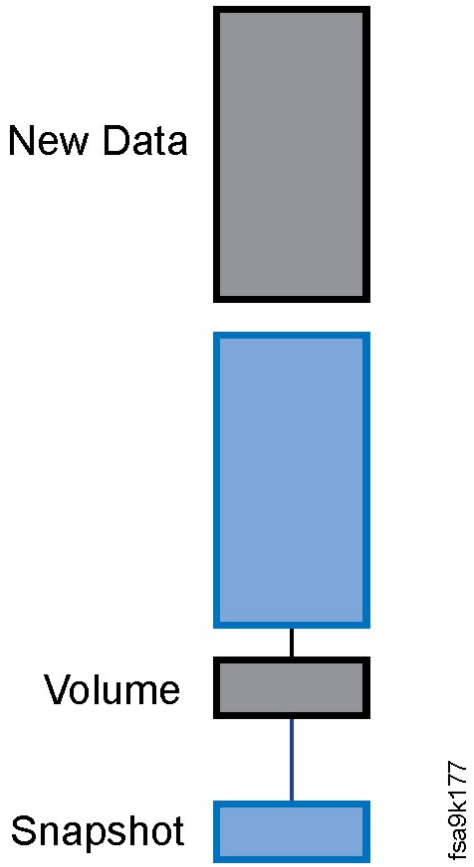


Figure 21. The Redirect-on-Write process: the new data is written

The snapshot points at the old data where the volume points at the new data (the data is regarded as new as it keep updating by I/Os).

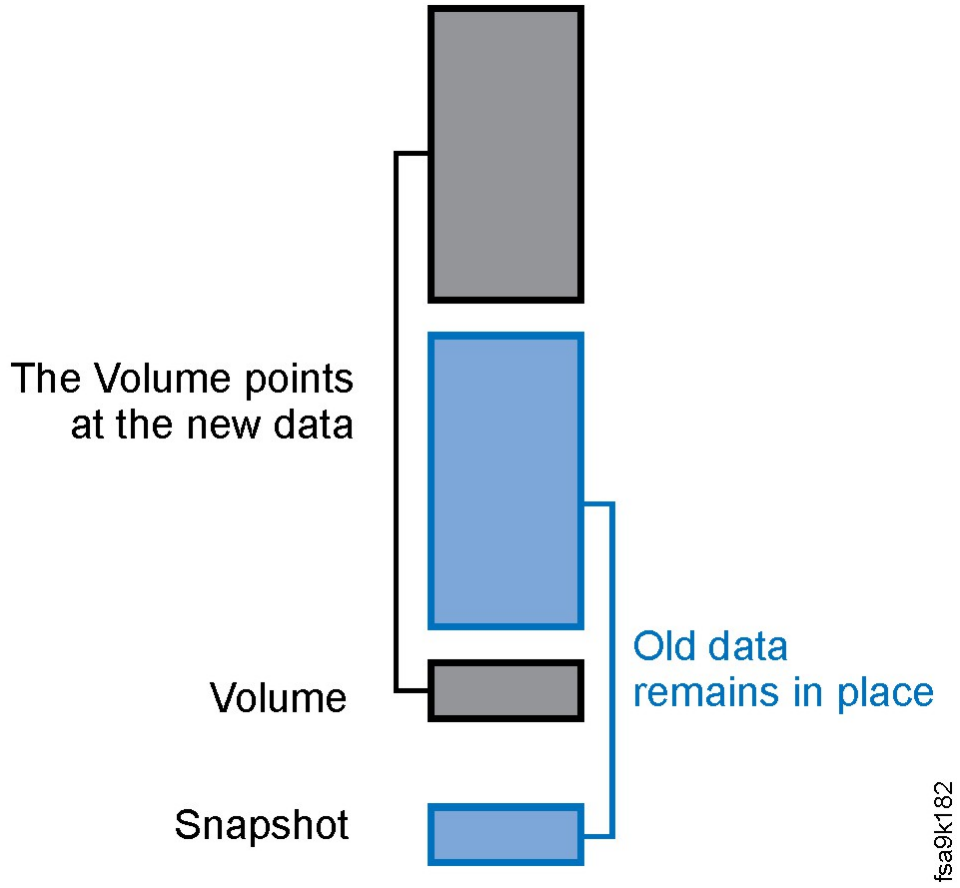


Figure 22. The Redirect-on-Write process: The snapshot points at the old data where the volume points at the new data

The metadata established at the beginning of the snapshot mechanism is independent of the size of the volume to be copied. This approach allows the user to achieve the following important goals:

Continuous backup
As snapshots are taken, backup copies of volumes are produced at frequencies that resemble those of *Continuous Data Protection* (CDP). Instant restoration of volumes to virtually any point in time is easily achieved in case of logical data corruption at both the volume level and the file level.

Productivity
The snapshot mechanism offers an instant and simple method for creating short or long-term copies of a volume for data mining, testing, and external backups.

Full volume copy

Full Volume Copy overwrites an existing volume, and at the time of its creation it is logically equivalent to the source volume.

After the copy is made, both volumes are independent of each other. Hosts can write to either one of them without affecting the other. This is somewhat similar to creating a writable (unlocked) snapshot, with the following differences and similarities:

Creation time and availability
Both Full Volume Copy and creating a snapshot happen almost instantly. Both the new snapshot and volume are immediately available to the host. This is because at the time of creation, both the source and the destination of the copy operation contain the exact same data and share the same physical storage.

Singularity of the copy operation
Full Volume Copy is implemented as a single copy operation into an existing volume, overriding its content and potentially its size. The existing target of a volume copy can be mapped to a host. From the host perspective, the content of the volume is changed within a single transaction. In contrast, creating a new writable snapshot creates a new object that has to be mapped to the host.

Space allocation
With Full Volume Copy, all the required space for the target volume is reserved at the time of the copy. If the storage pool that contains the target volume cannot allocate the required capacity, the operation fails and has no effect. This is unlike writable snapshots, which are different in nature.

Taking snapshots and mirroring the copied volume
The target of the Full Volume Copy is a Primary volume. This Primary volume can later be used as a source for taking a snapshot or creating a mirror. However, at the time of the copy, neither snapshots nor remote mirrors of the target volume are allowed.

Redirect-on-write implementation
With both Full Volume Copy and writable snapshots, while one volume is being changed, a redirect-on-write operation will ensure a split so that the other volume maintains the original data.

Performance
Unlike writable snapshots, with Full Volume Copy, the copying process is performed in the background even if no I/O operations are performed. Within a certain amount of time, the two volumes will use different copies of the data, even though they contain the same logical content. This means that the redirect-on-write overhead of writes occur only before the initial copy is complete. After this initial copy, there is no additional overhead.

Availability
Full Volume Copy can be performed with source and target volumes in different storage pools.

Restoring volumes and snapshots

The restoration operation provides the user with the ability to instantly recover the data of a Primary volume from any of its locked snapshots.

Restoring volumes

A volume can be restored from any of its snapshots, locked and unlocked. Performing the restoration replicates the selected snapshot onto the volume. As a result of this operation, the Primary volume is an exact replica of the snapshot that restored it.

All other snapshots, old and new, are left unchanged and can be used for further restore operations. A volume can even be restored from a snapshot that has been written to. The following figure shows a volume being restored from three

different snapshots.

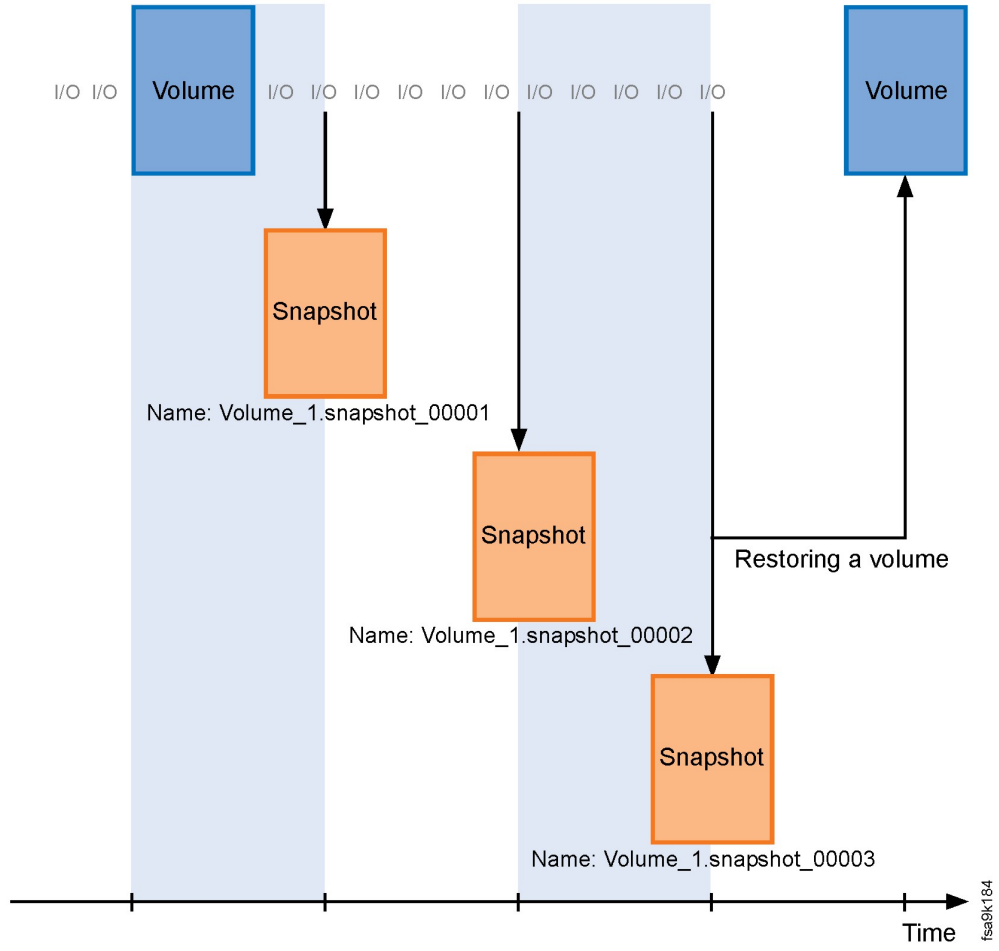


Figure 23. Restoring volumes

Restoring snapshots

The snapshot itself can also be restored from another snapshot. The restored snapshot retains its name and other attributes. From the host perspective, this restored snapshot is considered an instant replacement of all the snapshot content with other content.

The following figure shows a snapshot being restored from two different snapshots.

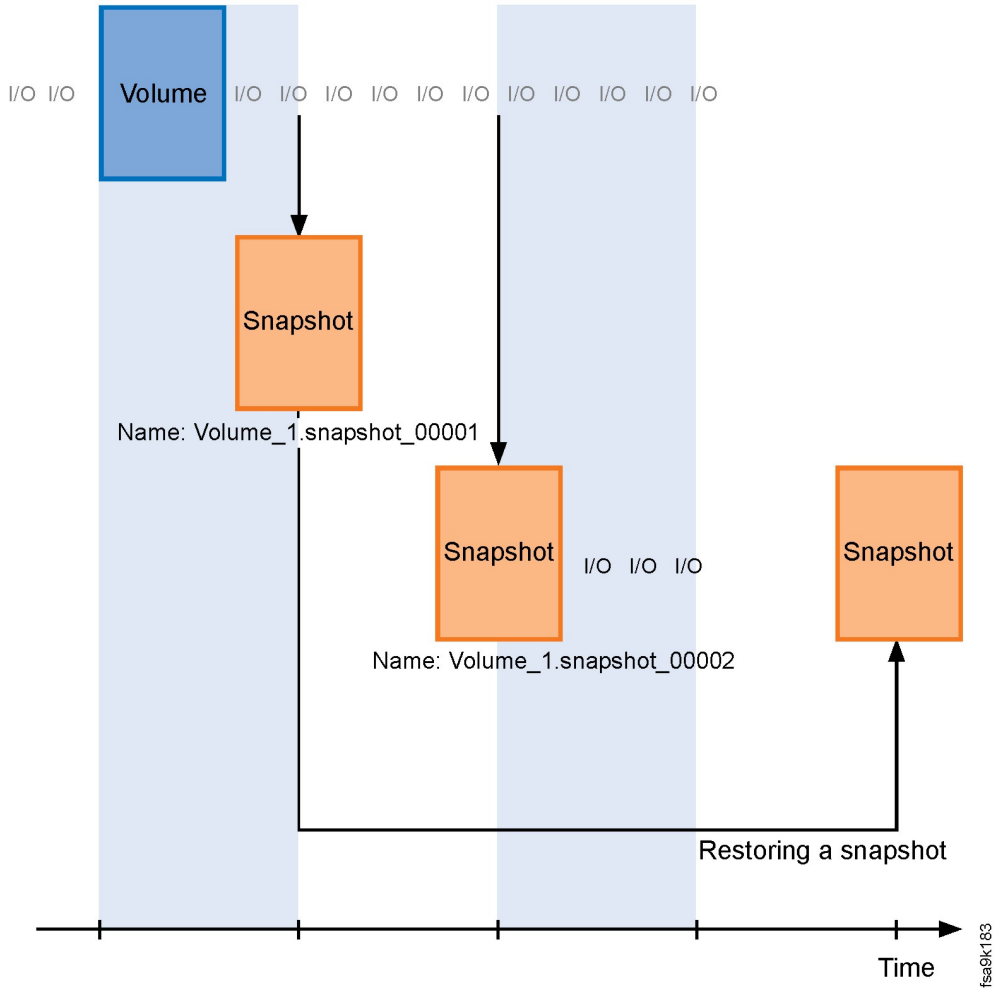


Figure 24. Restoring snapshots

Chapter 7. Storage pools

The allocatable capacity of the storage system can be partitioned into *storage pools*, where each volume belongs to a specific storage pool.

The concept of storage pools is purely administrative. Although the hardware resources within the storage system are virtualized in a global sense, the available capacity in the system can be administratively portioned into separate and independent storage pools. Essentially, storage pools function as a virtual way to effectively manage a related group of similarly provisioned logical volumes, consistency groups and their snapshots.

Because of the global deduplication in the storage system (see Chapter 2, "Flash-optimized data reduction," on page 9), the capacity assigned to a storage pool is virtual and not totally calculated on the physical or hard capacity of the system. A pool is still configured to be a specific size however the amount of capacity actually used will be dependent on the deduplication and compression savings realized based on the data written from the host system.

The minimum size of a storage pool is 103 GB of the allocation limit. The actual capacity is smaller, depending on the data saving ratio. The size of a storage pool can always be resized (grow or shrink). When decreasing the size of a pool, the only limit is that the new pool size cannot be less than the total size of configured volumes in a pool or not less than the amount of data written for an over allocated pool.

Storage pools as logical entities

A storage pool is a logical entity and is not associated with a specific flash enclosure or grid controller. All storage pools are equally spread over all flash enclosures and grid controllers in the system.

As a result, there are no limitations on the size of storage pools or on the associations between volumes and storage pools. For example:

- The size of a storage pool can be decreased, limited only by the space consumed by the volumes and snapshots in that storage pool.
- Volumes can be moved between storage pools without any limitations, as long as there is enough free space in the target storage pool.

All of the above transactions are accounting transactions, and do not impose any data copying from one disk drive to another. These transactions are completed instantly.

Moving volumes between storage pools

Volumes that are not in a consistency group (see Chapter 8, "Consistency groups," on page 43) can be moved between storage pools without any limitations (assuming adequate space in the new pool). Volumes that are part of a consistency group can be moved together as a group.

For a volume to be moved to a specific storage pool, there must be enough room for it to reside there. If a storage pool is not large enough, the storage pool must be resized, or other volumes must be moved out to make room for the new volume.

A volume and all its snapshots always belong to the same storage pool. Moving a volume between storage pools automatically moves all its snapshots together with the volume.

Protecting snapshots at the storage pool level

Snapshot space must be reserved during pool definition if any of the volumes in the pool will be duplicated with snapshots. This is included as part of the usable capacity in the storage pool. A pool can be resized at a later time to add or remove snapshot space as needed.

Snapshots that participates in the mirroring process can be protected in case of pool space depletion.

This is done by attributing both snapshots (or snapshot groups) and the storage pool with a deletion priority. The snapshots are attributed with a deletion priority between 0 to 4 and the storage pool is configured to disregard snapshots whose priority is above a specific value.

Snapshots with a lower delete priority (higher number) than the configured value might be deleted by the system whenever the pool space depletion mechanism implies so, thus protecting snapshots with a priority equal or higher to this value.

Chapter 8. Consistency groups

A *consistency group* is a group of volumes of which a snapshot can be made at the same point in time, therefore ensuring a consistent image of all volumes within the group at that time.

The concept of a consistency group is common among storage systems in which it is necessary to perform concurrent operations collectively across a set of volumes so that the result of the operation preserves the consistency among volumes. For example, effective storage management activities for applications that span multiple volumes, or creating point-in-time backups, is not possible without first employing consistency groups.

The consistency between the volumes in the group is important for maintaining data integrity from the application perspective. By first grouping the application volumes into a consistency group, it is possible to later capture a consistent state of all volumes within that group at a specified point-in-time using a special snapshot command for consistency groups.

Consistency groups can be used to take simultaneous snapshots of multiple volumes, thus ensuring consistent copies of a group of volumes.

A consistency group is also an administrative unit that facilitates simultaneous snapshots of multiple volumes, mirroring of volume groups, and administration of volume sets.

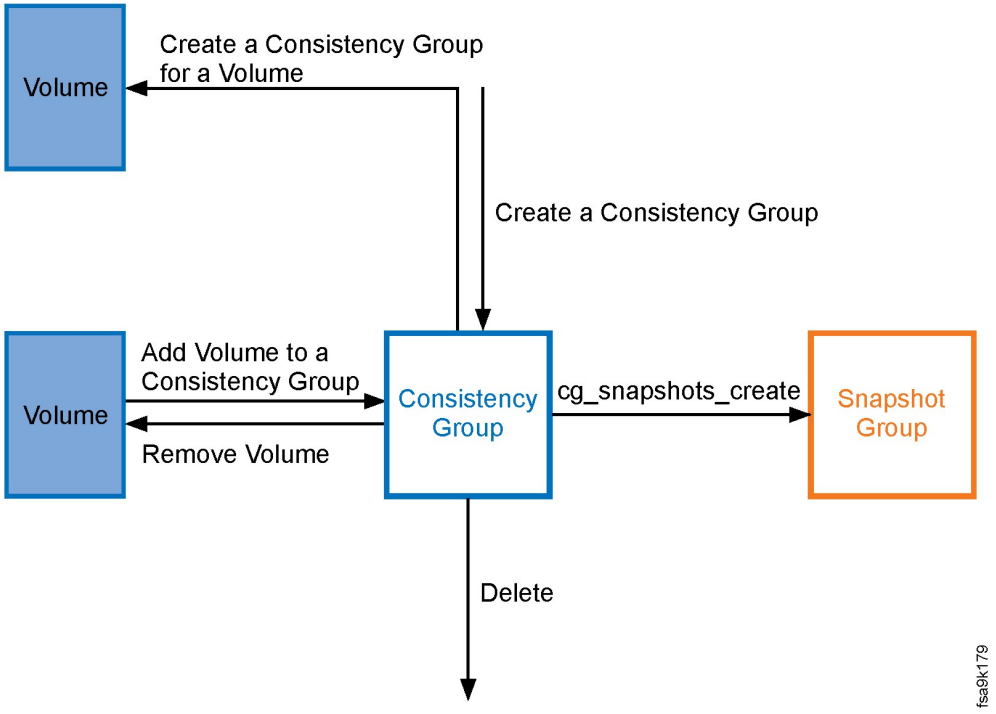


Figure 25. Consistency group creation and options

All volumes in a consistency group must belong to the same storage pool.

Snapshot of a consistency group

Taking a snapshot for an entire consistency group means that a snapshot is taken for each volume of the consistency group at the same point-in-time.

These snapshots are grouped together to represent the volumes of the consistency group at a specific point in time.

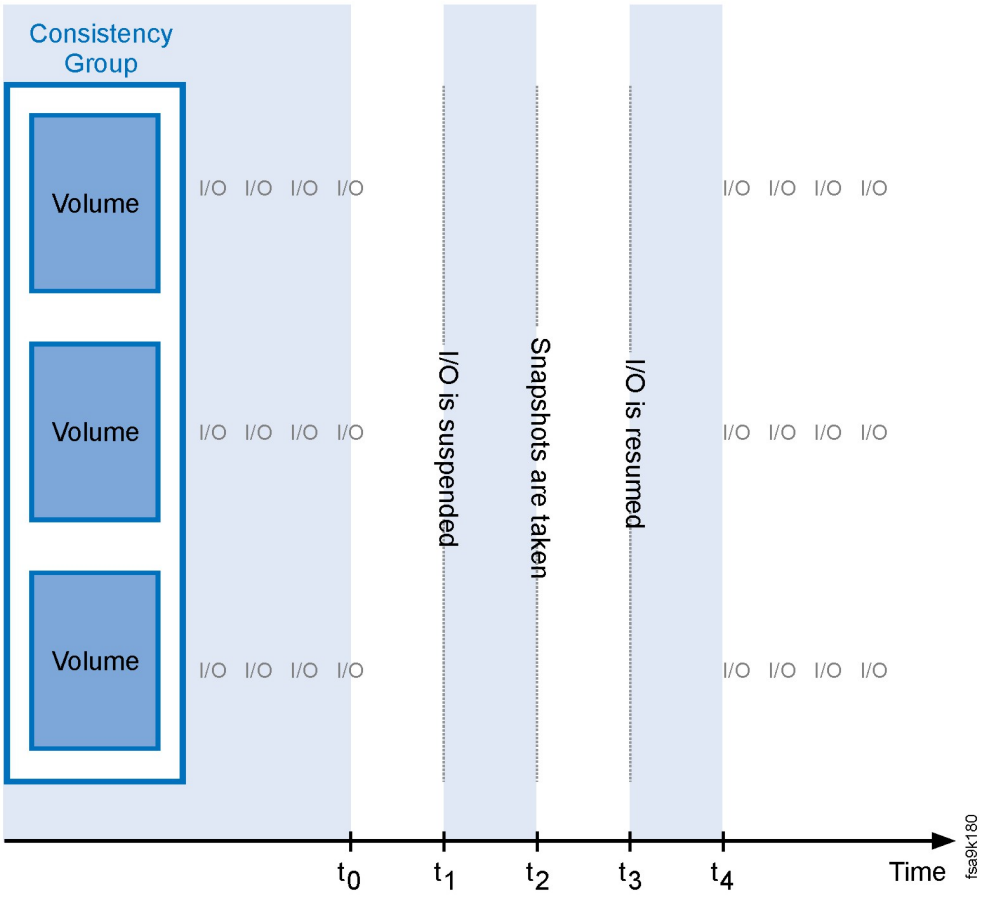


Figure 26. A snapshot is taken for each volume of the consistency group

As shown in the above figure, a snapshot is taken for each of the consistency group's volumes in the following order:

Time = t_0
Prior to taking the snapshots, all volumes in the consistency group are active and being read from and written to.

Time = t_1
When the command to snapshot the consistency group is issued, I/O is suspended .

Time = t_2
Snapshots are taken at the same point in time.

Time = t_3
I/O is resumed and the volumes continue their normal work.

Time = t_4
After the snapshots are taken, the volumes resume active state and continue to be read from and written to.

Most snapshot operations can be applied to each snapshot in a grouping, known as a *snapshot set*. The following items are characteristics of a snapshot set:

- A snapshot set can be locked or unlocked. When you lock or unlock a snapshot set, all snapshots in the set are locked or unlocked.
- A snapshot set can be duplicated.
- A snapshot set can be deleted. When a snapshot set is deleted, all snapshots in the set are also deleted.

A snapshot set can be disbanded which makes all the snapshots in the set independent snapshots that can be handled individually. The snapshot set itself is deleted, but the individual snapshots are not.

Consistency group snapshot lifecycle

Most snapshot operations can be applied to snapshot groups, where the operation affects every snapshot in the group.

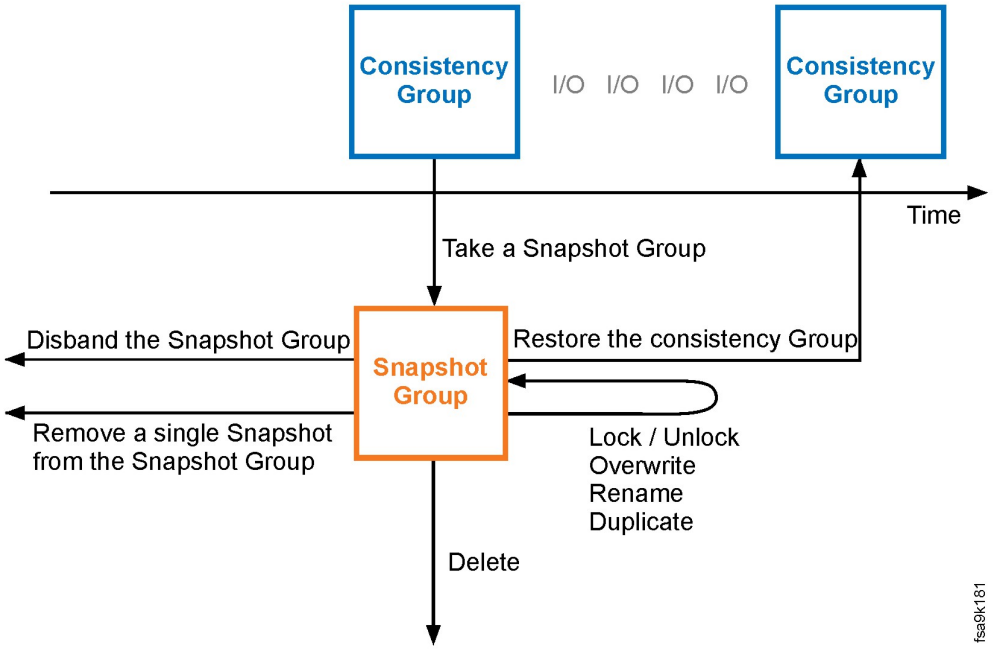


Figure 27. Most snapshot operations can be applied to snapshot groups

Taking a snapshot group
Creates a snapshot group.

Restoring consistency group from a snapshot group
The main purpose of the snapshot group is the ability to restore the entire consistency group at once, ensuring that all volumes are synchronized to the same point in time.

Restoring a consistency group is a single action in which every volume that belongs to the consistency group is restored from a corresponding

snapshot that belongs to an associated snapshot group. Not only does the snapshot group have a matching snapshot for each of the volumes, all of the snapshots have the same time stamp. The restored consistency group contains a consistent image of its volumes as they were at a specific point in time.

Note: A consistency group can only be restored from a snapshot group that has a snapshot for each of the volumes. If either the consistency group or the snapshot group has changed after the snapshot group is taken, the restore action does not work.

Listing a snapshot group
This command lists snapshot groups with their consistency groups and the time the snapshots were taken.

Note: All snapshots within a snapshot group are taken at the same time.

Lock and unlock
Similar to unlocking and locking an individual snapshot, the snapshot group can be rendered writable, and then be written to. A snapshot group that is unlocked cannot be further used for restoring the consistency group, even if it is locked again.

The snapshot group can be locked again. At this stage, it cannot be used to restore the Primary consistency group. In this situation, the snapshot group functions like a consistency group of its own.

Overwrite
The snapshot group can be overwritten by another snapshot group.

Rename
The snapshot group can be renamed.

Restricted names
Do not prefix the snapshot group's name with any of the following strings:

1. `most_recent`
2. `last_replicated`

Duplicate
The snapshot group can be duplicated, thus creating another snapshot group for the same consistency group with the time stamp of the first snapshot group.

Disbanding a snapshot group
The snapshots that comprise the snapshot group are each related to its volume. Although the snapshot group can be rendered inappropriate for restoring the consistency group, the snapshots that comprise it are still attached to their volumes. Disbanding the snapshot group detaches all snapshots from this snapshot group but maintains their individual connections to their volumes. These individual snapshots cannot restore the consistency group, but they can restore its volumes individually.

Changing the snapshot group deletion priority
Manually sets the deletion priority of the snapshot group.

Deleting the snapshot group
Deletes the snapshot group along with its snapshots.

Chapter 9. Quality of Service (QoS) performance classes

The Quality of Service (QoS) feature allows the storage system to deliver different service levels to hosts that are connected to the same storage system.

The QoS feature favors performance of critical business applications that run concurrently with noncritical applications. Because the flash disk and cache are shared among all applications and all hosts are attached to the same resources, division of these resources among both critical and noncritical applications might have an unintended adverse performance effect on critical applications.

QoS can address this by limiting the rate, based on bandwidth and IOPS, for non-critical applications. Limiting performance resources for non-critical applications means that the remaining resources are available without limitation for the business-critical applications.

The QoS feature is managed through the definition of performance classes and then associating hosts with a performance class. It also applies to storage pools (see Chapter 7, "Storage pools," on page 41) and storage domains (see "Working with multi-tenancy" on page 117). Each performance class is now implicitly one of two types: host type or pool/domain type.

The QoS feature possibilities can be summarized as follows:

- Up to 512 performance classes are configurable.
- QoS is applicable to host, domain, pool, volume, and restricted combinations of these entities. For example, hosts cannot be specified for a performance class that already contains a domain or pool.
- Limits can be defined *Per Interface*.
- Limits are specified as IOPS or bandwidth.
- Limit calculation is based on preferred practices for setup and zoning.

  The limited I/O processes are expected to always come through all active interface nodes (equal to active interface modules).

Note: For better performance, create one performance class per domain and one performance class per storage pool.

Max bandwidth limit attribute

The host rate limitation group has a max bandwidth limit attribute, which is the number of blocks per second. This number could be either:

- A value between **`min_rate_limit_bandwidth_blocks_per_sec`** and **`max_rate_limit_bandwidth_blocks_per_sec`** (both are available from the storage system's configuration).
- Zero (0) for unlimited bandwidth.

Chapter 10. Connectivity with hosts

The storage system connectivity is provided through the following interfaces:

- 16-Gigabit (16 Gbps) Fibre Channel (FC) interfaces for host-based I/O using the FC protocol (FCP) over Fibre Channel networks
- 10-Gigabit Ethernet (10 Gbps) interfaces for host-based I/O using the iSCSI protocol over IP or Ethernet networks
- Gigabit Ethernet for management (GUI or CLI) connectivity
- Ethernet interface for incoming-only traffic of technician maintenance operations, provided through the maintenance module

The total number and type of connectivity ports depends on the number of grid controllers in the storage system configuration. Systems can be ordered with iSCSI-only interfaces.

IBM FlashSystem A9000 and A9000R storage systems with enhanced grid controllers feature FC-NVMe ready adapters (see the Fibre Channel NVMe connectivity hardware announcement). NVM Express (NVMe) allows servers to leverage the native parallelism of today's SSD offerings, reduces overall I/O overhead, and increases bandwidth. FC-NVMe enables NVMe over a Fibre Channel (FC) network fabric, thus combining the benefits of all-flash SAN storage with NVMe performance over existing infrastructure. A system is FC-NVMe ready if it requires a future software update to provide full FC-NVMe support.

The following subsections provide information about different connectivity aspects.

IP and Ethernet connectivity

The following subtopics provide a basic explanation of the various Ethernet and IP connectivity interfaces that can be used in various configurations.

Ethernet ports

10-Gigabit Ethernet iSCSI ports
These ports are used for iSCSI over IP or Ethernet services. A fully equipped rack is configured with six Ethernet ports for iSCSI service. These ports should connect to the user's IP network and provide connectivity to the iSCSI hosts. The iSCSI ports can also accept management connections.

Gigabit Ethernet management ports
These ports are dedicated for CLI and GUI communications, as well as for outgoing SNMP and SMTP connections.

Field technician ports
These Ethernet ports are used for incoming management traffic only. The ports are utilized only for the field technician's laptop computer and must not be connected to the user's IP network.

Optional IPv6 connectivity for management traffic

The storage system supports optional IPv6 through stateless auto-configuration and full IPsec (IKE2, transport, and tunnel mode) for the management ports. IPv6 is not supported for the iSCSI or field technician ports.

When IPv6 is enabled, stateless auto-configuration is automatically enabled as well, and the system interfaces are getting ready to work with IPv6. Accordingly, when looking for DNS addresses, the system also looks for AAAA entries. In addition, each IP interface in the system may have several IP addresses: static IPv4 address, static IPv6 address, and the stateless configuration link and site local IPv6 addresses. Where multiple IPv6 static addresses are assigned for each interface, the system supports only one address per interface.

Programs that are using connections on the management and VPN ports must support IPv6 addresses.

Management connectivity

Management connectivity is used for the following functions:

- Issuing CLI commands through the XCLI utility.
- Controlling the storage system through the management GUI (IBM Hyper-Scale Manager Hub).
- Sending e-mail notification messages and SNMP traps about event alerts.

To ensure management redundancy in case of module failure, the storage system management function is accessible from two different IP addresses in IBM FlashSystem A9000, and three in IBM FlashSystem A9000R. Each of the three IP addresses is handled by a different hardware module. The various IP addresses are transparent to the user and management functions can be performed through any of the IP addresses. These addresses can be accessed simultaneously by multiple clients. Users only need to configure the set of management IP addresses that are defined for the specific system.

Note: All management IP interfaces must be connected to the same subnet and use the same network mask, gateway, and MTU.

The management connectivity allows users to manage the system from both the CLI and GUI. Accordingly, both can be configured to manage the system through iSCSI IP interfaces. Both CLI and GUI management is run over TCP port 7778. With all traffic encrypted through the Secure Sockets Layer (SSL) protocol.

System-initiated IP communication

The storage system can also initiate IP communications to send event alerts as necessary. Two types of system-initiated IP communications exist:

Sending e-mail notifications through the SMTP protocol
E-mails are used for both e-mail notifications and for SMS notifications through the SMTP to SMS gateways.

Sending SNMP traps

Note: SMPT and SNMP communications can be initiated from any of the three IP addresses. This is different from the CLI and GUI, which are user initiated. Accordingly, it is important to configure all three IP interfaces and to verify that they have network connectivity.

Host system attachment

Hosts of various operating systems can be attached to the storage system over iSCSI or Fibre Channel connections.

The following subtopics provide information about different host system attachment aspects.

Balanced traffic and no single point of failure

Although the storage system distributes the traffic across all system modules, the storage administrator is responsible for ensuring that host I/O operations are equally distributed among the different interface modules.

The workload balance should be monitored and reviewed when host traffic patterns change. The storage system does not automatically balance incoming host traffic.

The storage administrator is responsible for ensuring that host connections are made redundantly in such a way that a single failure, such as in a module or HBA, will not cause all paths to the machine to fail. In addition, the storage administrator is responsible for making sure that the host workload is adequately spread across the different connections and interface modules.

Attaching volumes to hosts

While the storage system identifies volumes and snapshots by name (see Chapter 6, "Volumes and snapshots," on page 27), hosts identify volumes and snapshots according to their logical unit number (LUN).

A *LUN* is an integer that is used when attaching a system's volume to a registered host. Each host can access some or all of the volumes and snapshots on the storage system, up to a set maximum. Each accessed volume or snapshot is identified by the host through a LUN.

For each host, a LUN identifies a single volume or snapshot. However, different hosts can use the same LUN to access different volumes or snapshots.

To facilitate the procedure of volume attachment, the IBM Storage Host Attachment Kit (HAK) can be used. The software kit provides a set of command-line interface (CLI) tools that help host administrators perform different host-side tasks, such as: detect any physically connected storage system, detect systems and volumes, obtain detailed host information, define the host on the storage system, run diagnostics, and apply best practice native multipath connectivity configuration on the host.

Important: LUN0 can be mapped to a volume like any other LUN. However, when no volume is mapped to LUN0, the HAK uses it to discover the LUN array. Accordingly, LUN0 should not be used as a normal LUN.

Multipathing

The system's host connectivity interfaces utilize multipathing access algorithms. When a host connects to the system through several independent ports, each volume can be accessed directly through any of the host connectivity interfaces, and no further interaction is required. This allows using more than one physical path for transferring data between the host and the storage system.

CHAP authentication of iSCSI hosts

Hosts that access the storage system can be authenticated by using the Challenge-Handshake Authentication Protocol (CHAP).

When CHAP support is enabled, hosts are securely authenticated by the storage system. This increases overall system security by verifying that only authenticated parties are involved in host-storage interactions.

Definitions

CHAP authentication
An authentication process of an iSCSI initiator by a target through comparing a secret hash that the initiator submits with a computed hash of that initiator's secret which is stored on the target.

Initiator
The host.

One-way (unidirectional CHAP)
CHAP authentication where initiators are authenticated by the target, but not vice-versa.

Supported configurations

CHAP authentication type
One-way (unidirectional) authentication mode, meaning that the Initiator (host) has to be authenticated by the storage system.

MD5 CHAP authentication utilizes the MD5 hashing algorithm.

Access scope
CHAP-authenticated initiators are granted access to the storage system by defining mapping that may restrict access to some volumes.

Authentication modes

The following authentication modes are supported:

None (default)
In this mode, an initiator is not authenticated by the storage system.

CHAP (one-way)
In this mode, an initiator is authenticated by the storage system based on the pertinent initiator's submitted hash, which is compared to the hash computed from the initiator's secret stored on the IBM XIV Storage System.

Changing the authentication mode from None to CHAP requires an authentication of the host. Changing the mode from CHAP to None does not require an authentication.

Complying with RFC 3720

The CHAP authentication procedure complies with the CHAP requirements as defined in IETF RFC 3720 (http://tools.ietf.org/html/rfc3720).

Secret length
The secret key has to be between 96 bits and 128 bits.

Initiator secret uniqueness
Upon defining or updating an initiator (host) secret, the system compares

the entered secret's hash with existing secrets stored by the system and determines whether the secret is unique. If it is not unique, the system presents a warning to the user, but does not prevent the command from completing successfully.

Clustering hosts into LUN maps

To enhance the management of hosts, the storage system allows clustering hosts together while the clustered hosts are provided with identical mappings.

The mapping of volumes to LUN identifiers is defined per cluster and applies to all of the hosts in the cluster.

Adding and removing hosts to a cluster are done as follows:

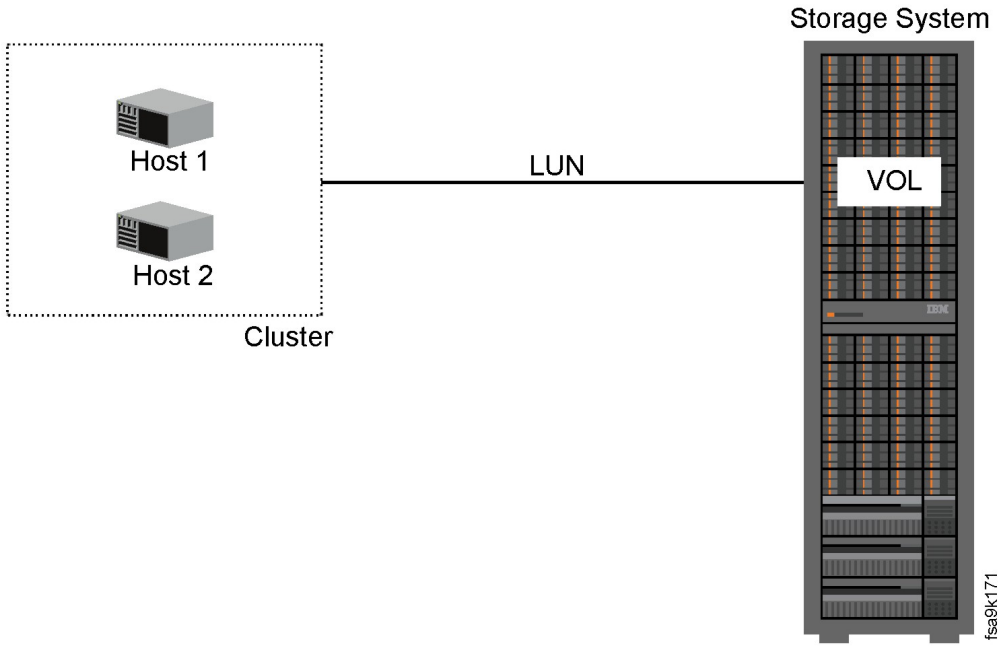


Figure 28. A volume, a LUN, and cluster hosts

Adding a host to a cluster

Adding a host to a cluster is a straightforward action in which a host is added to a cluster and is connected to a LUN. This results in either:

- Changing the host mapping to the cluster mapping.
- Changing the cluster mapping to be identical to the mapping of the newly added host.

Removing a host from a cluster

When a host is removed from the cluster:

- The removed host mapping remains identical to the LUN mapping of the cluster.
- The host mapping definitions do not revert to the original mapping (the mapping that was in effect before the host was added to the cluster).
- The host LUN mapping can be changed.

The following restrictions are applicable:

- The storage system defines the same mapping to all of the hosts of the same cluster. No hierarchy of clusters is maintained.
- A volume cannot be mapped to an already mapped LUN.

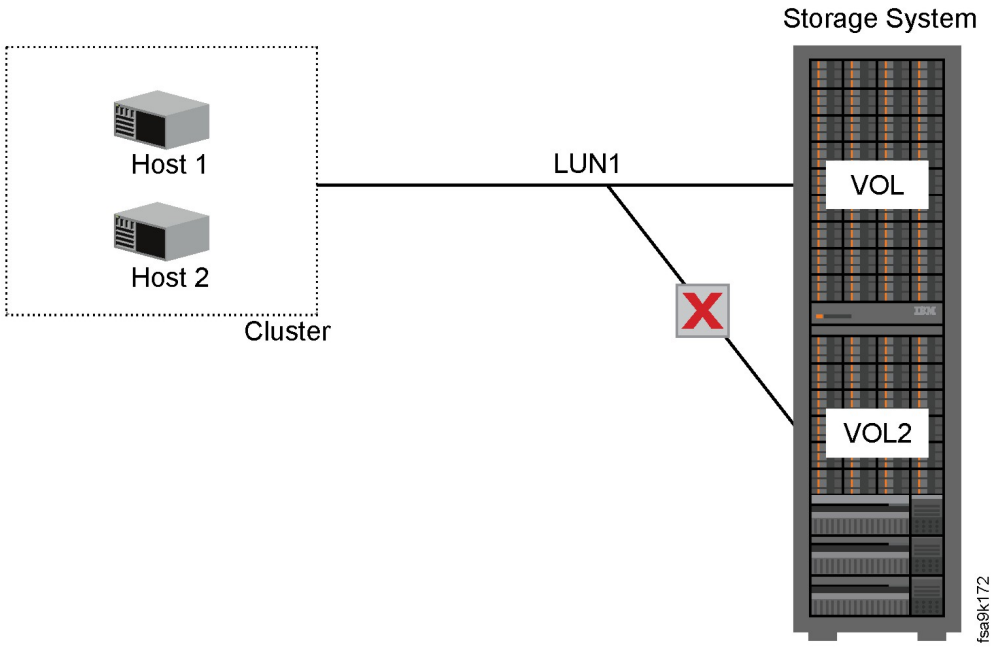


Figure 29. Volume that cannot be mapped to an already mapped LUN

- A mapped volume cannot be mapped to another LUN.

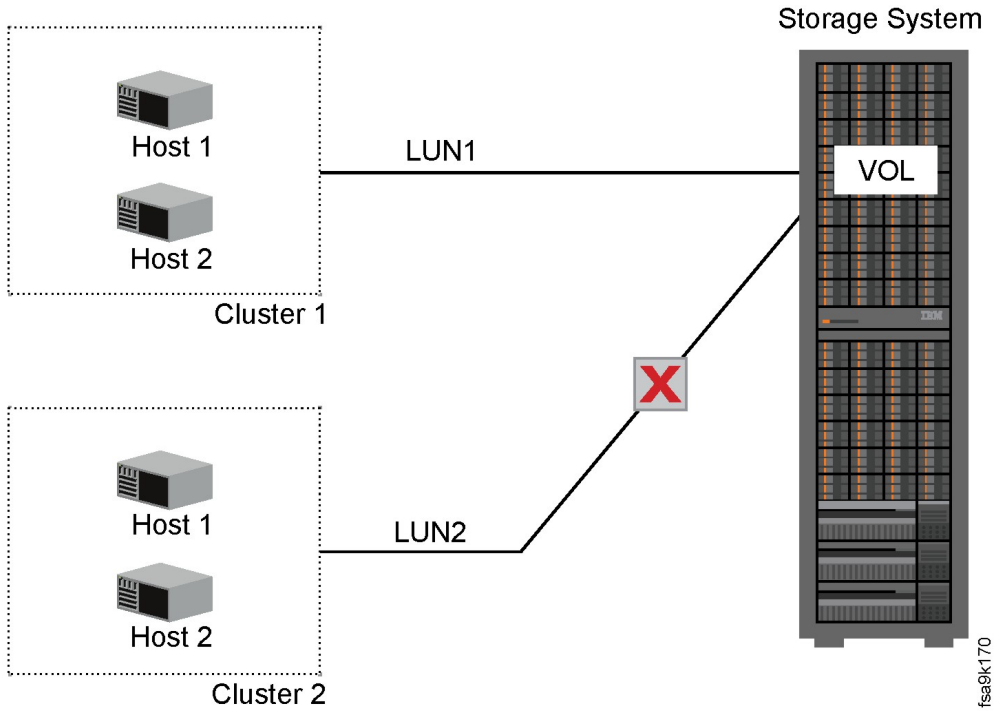


Figure 30. Mapped volume that cannot be mapped to another LUN

Chapter 11. Synchronous remote mirroring

Remote mirroring allows replication of data between two geographically remote sites, allowing full data recovery from the remote site in different disaster scenarios.

The process of ensuring that both storage systems contain identical data at all times is called *remote mirroring*. Remote mirroring can be established between two remote storage systems to provide data protection for the following types of site disasters:

Local site failure
When a disaster occurs at a certain site, the remote site takes over and maintains full service to the hosts connected to the original site. The mirroring is resumed after the failing site recovers.

Split-brain scenario
After a communication loss between the two sites, each site maintains full service to the hosts. After the connection is resumed and the link (mirror) is established, the sites complement each other's data to regain full synchronization.

For the information on the cross-system use of synchronous remote mirroring, see the feature availability matrix.

Remote mirroring basic concepts

Synchronous remote mirroring provides continuous availability of critical information in the case of a disaster scenario.

A typical remote mirroring configuration involves the following two sites:

Primary site
The location of the primary storage system.

A local site that contains both the data and the active servers.

Servers may simultaneously perform primary or secondary roles with respect to their hosts. As a result, a server at one site can be the primary storage system for a specific application, while simultaneously being the secondary storage system for another application.

Secondary site
The location of the secondary backup storage system.

A remote site that contains a copy of the data and standby servers. Following a disaster at the primary site, the servers at the secondary site become active and start using the copy of the data.

Primary volume
The volume which is mirrored. The Primary volume is usually located at the primary site.

Secondary volume
The volume to which the Primary volume is mirrored. The Secondary volume is usually located at the secondary site.

Synchronous remote mirroring is performed during each write operation. The write operation issued by a host is applied to both the primary and the secondary storage systems.

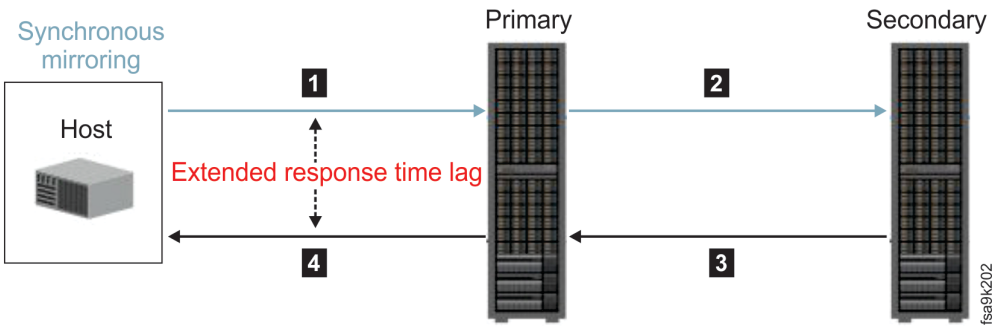


Figure 31. Synchronous remote mirroring scheme

Note: When using remote mirroring with FlashSystem A9000 or A9000R, data is transferred over the mirror connectivity in uncompressed format. The data is deduplicated and compressed again after it reaches the remote system.

When a volume is mirrored, reading is performed from the Primary volume, while writing is performed on both the Primary and the Secondary volumes, as previously described.

Synchronous mirroring operations

Remote mirroring operations involve *configuration*, *initialization*, *ongoing operation*, *handling of communication failures*, and *role switching*.

The following list describes the remote mirroring operations:

Configuration
Configuration is the act of defining Primary and Secondary volumes for a mirror relation.

Initialization
Remote mirroring operations begin with a Primary volume that contains data and a new Secondary volume. Next, data is copied from the Primary volume to the Secondary volume. This process is called *initialization*. Initialization is performed once in the lifetime of a remote mirroring coupling. After it is successfully completed, both volumes are synchronized.

Ongoing operation
After the *initialization* process is complete, remote mirroring is activated. During this activity, all data is written to the Primary volume and to the Secondary volume. The write operation is complete after an acknowledgment is received from the Secondary volume. At any point, the Primary and Secondary volumes contain identical data except for any unacknowledged (pending) writes.

Handling of communication failures
Communication between sites may break. In this case, the primary site continues its function and updates the secondary site after communication resumes. This process is called *synchronization*.

Role switching
When needed, a volume can change its role from Primary to Secondary or vice versa, either as a result of a disaster at the primary site, maintenance operations, or intentionally, to test the disaster recovery procedures.

Using snapshots in synchronous mirroring

The storage system uses snapshots to identify inconsistencies that may arise between updates.

If the link between volumes is disrupted or if the mirroring is deactivated, the Primary volume continues accepting host writes, but does not replicate the writes onto the Secondary volume. After the mirroring is restored and activated, the system takes a snapshot of the Secondary volume, which represents the data that is known to be mirrored. This snapshot is called the *last-consistent snapshot*. Only then more recent writes to the Primary volume are replicated to the Secondary volume.

The last-consistent snapshot is automatically deleted after the resynchronization is complete for all mirrors on the same target. However, if the Secondary volume role is changed to Primary during resynchronization, the last-consistent snapshot will not be deleted.

Synchronous mirroring configuration and activation options

The remote mirroring configuration process involves configuring volumes and volume pairs.

Volume configuration

The following concepts are to be configured for volumes and the relations between them:

The *volume role* is the current function of the volume. The following volume roles are available:

None The volume is created using normal volume creation procedures and is not mirrored.

Primary
The volume is directly written to by the host.

Secondary
A backup to the Primary volume.

Data can be read from the Secondary volume by a host. Data cannot be written to the Secondary volume by any host.

Mixed configuration
In some cases, the volumes on a single storage system can be defined in a *mixed configuration*. For example, a storage system can contain volumes whose role is defined as Primary, as well as volumes whose role is defined as Secondary. In addition, some volumes might not be involved in a remote mirroring coupling at all.

Configuration error
In some cases, configuration on both sides might be changed in a non-compatible way. This is defined as a *configuration error*. For example, switching the role of only one side when communication is down causes a

configuration error when connection resumes, because each side is configured as a Primary or Secondary.

Coupling activation

When a pair of volumes point to each other, it is referred to as a *coupling*. In a *coupling relationship*, two volumes, referred to as *peers*, participate in a remote mirroring system with the Secondary peer serving as the backup for the Primary peer. The coupling configuration is identical for both Primary and Secondary volumes.

Remote mirroring can be manually activated and deactivated per coupling. When activated, the coupling is in *Active* mode. When deactivated, the coupling is in *Standby* mode.

These modes have the following functions:

Active Remote mirroring is functioning and the data is replicated.

Standby
Remote mirroring is deactivated. The data is not replicated to the Secondary volume.

Standby mode is used mainly when maintenance is performed on the secondary site or during communication failures between the sites. In this mode, the Primary volumes will not generate mirroring-failure alerts.

The coupling lifecycle has the following characteristics:

- When a coupling is created, it is always initially in *Standby* mode.
- Only a coupling in *Standby* mode can be deleted.

Supported network configurations

Synchronous mirroring supports the following network configurations:

- Either Fibre Channel (FC) or iSCSI connectivity can be used for replication, regardless of the connectivity that is used by the host to access the Primary volume.
- The remote system must be defined in the *remote target connectivity* definitions.
- All the volumes that belong to the same consistency group must reside on the same remote system.
- Primary and Secondary volumes must have exactly the same size.

Synchronous mirroring statuses

The status of a synchronous remote mirroring volume depends on the communication link and on the coupling between the Primary volume and the Secondary volume.

The following table lists the different statuses of a synchronous remote mirroring volume during remote mirroring operations.

Table 5. Synchronous mirroring statuses

Entity	Status type	Possible status values	Description
Link	Operational status	• Up • Down	Specifies if the communications link is up or down. The link status of the Primary volume is also the link status of the Secondary volume.
Coupling	Operational status	• Operational • Non-operational	Specifies if remote mirroring is working. To be operational, the link status must be up and the coupling must be activated. If the link is down or if the remote mirroring feature is in Standby mode, the status is Non-operational.
	Synchronization status	• Initialization • Synchronized • Unsynchronized • Consistent • Inconsistent	For detailed description of each status, see "Synchronization status" below.
	Last-secondary timestamp	Point-in-time date	Timestamp for when the secondary volume was last synchronized.
	Synchronization progress	Synchronization status	The relative portion of data remaining to be synchronized between the Primary and Secondary volumes due to non-operational coupling.
	Secondary-locked	Boolean	If the Secondary volume is locked for writing due to lack of space, the *Secondary-locked* status is *true*. This may occur during the synchronization process, when there is not enough space for the last-consistent snapshot. Otherwise, the *Secondary-locked* status is *false*.
	Configuration error	Boolean	If the configuration of the Primary and Secondary volumes is inconsistent., the *Configuration error* status is *true*.

Synchronization status

The synchronization status reflects the consistency of the data between the Primary and Secondary volumes.

Because remote mirroring is for ensuring that the Secondary volume is an identical copy of the Primary volume, this status indicates whether this objective is currently attained.

The possible synchronization statuses for the Primary volume are:

Initialization
The first step in remote mirroring is to create a copy of the data from the Primary volume to the Secondary volume. During this step, the coupling status remains *Initialization*.

Synchronized (Primary volume only)
This status indicates that all data that was written to the Primary volume and acknowledged has also been written to the Secondary volume. Ideally, the Primary and Secondary volumes should always be synchronized. This does not imply that the two volumes are identical because at any time there might be a limited amount of data that was written to one volume, but was not yet acknowledged by the Secondary volume. These are also known as *pending writes*.

Unsynchronized (Primary volume only)

After a volume has completed the *Initialization* stage and achieved the *Synchronized* status, a volume can become unsynchronized. This occurs when it is not known whether all the data that was written to the Primary volume was also written to the Secondary volume. This status occurs in the following cases:

- **Communications link is down** – As a result of the communication link going down, some data might have been written to the Primary volume, but was not yet replicated to the Secondary volume.
- **Secondary system is down** – This is similar to communication link errors because in this state, the primary system is updated while the secondary system is not.
- **Remote mirroring is deactivated** – As a result of the remote mirroring deactivation, some data might have been written to the Primary volume and not to the Secondary volume.

Consistent
The Secondary volume is an identical copy of the Primary volume.

Inconsistent
There is a discrepancy between the data on the Primary and Secondary volumes.

It is always possible to reestablish the Synchronized status when the link is reestablished or the remote mirroring feature is reactivated, no matter what was the reason for the *Unsynchronized* status.

Because all updates to the Primary volume that are not written to the Secondary volume are recorded, these updates are written to the Secondary volume. The synchronization status remains *Unsynchronized* from the time that the coupling is not operational until the synchronization process is completed successfully.

Last-secondary timestamp

A timestamp is taken when the coupling between the Primary and Secondary volumes becomes non-operational.

This time stamp specifies the last time that the Secondary volume was consistent with the Primary volume. This status has no meaning if the coupling's synchronization state is still *Initialization*.

For synchronized coupling, this timestamp specifies the current time. Most importantly, for an unsynchronized coupling, this timestamp denotes the time when the coupling became non-operational.

The timestamp is returned to current only after the coupling is operational and the Primary and Secondary volumes are synchronized.

Synchronization progress

During the synchronization process, when the Secondary volumes are being updated with previously written data, the volumes are given a dynamic synchronization process status.

This status comprises the following sub-statuses:

Size to complete
The size of data that requires synchronization.

Part to synchronize
The size to synchronize divided by the maximum size-to-synchronize since the last time the synchronization process started. For coupling initialization, the size-to-synchronize is divided by the volume size.

Time to synchronize
Time estimation that is required to complete the synchronization process and achieve synchronization, based on past rate.

Secondary-locked error status

When synchronization is in progress, there is a period in which the Secondary volume is not consistent with the Primary volume. While in this state, the Secondary volume maintains a last-consistent snapshot. Provided that every I/O operation requires a copy-on-write partition, this may result in insufficient space and, consequently, in the failure of I/O operations to the Secondary volume.

Whenever I/O operations to the Secondary volume fail due to insufficient space, all couplings in the system are set to the *Secondary-locked* status and become non-operational. The administrator is notified of a critical event, and can free space on the system containing the Secondary volume.

Synchronous mirroring role switchover and role change

When role switchover occurs, the Primary volume becomes the Secondary volume, and the Secondary volume becomes the Primary volume.

Role switching can occur when the synchronous remote mirroring function is either operational or not operational, as described in the following sections.

Role switchover when synchronous mirroring is operational

When the remote mirroring function is operational, role switching between Primary and Secondary volumes can be initiated from the management GUI or CLI.

There are two typical reasons for performing a switchover when communication between the volumes exists:

Drills Drills can be performed on a regular basis to test the functioning of the

secondary site. In a drill, an administrator simulates a disaster and tests that all procedures are operating smoothly.

Scheduled maintenance
To perform maintenance at the primary site, switch operations to the secondary site on the day before the maintenance. This can be done as a preemptive measure when a primary site problem is known to occur.

The CLI command that performs the role switchover must be run on the Primary volume. The switchover cannot be performed if the Primary and Secondary volumes are not synchronized.

Role change when synchronous mirroring is not operational

A more complex situation for role switching is when there is no communication between the two sites, either because of a network malfunction, or because the primary site is no longer operational.

The CLI command for this scenario is **mirror_change_role**. Because there is no communication between the two sites, the command should be issued on both sites concurrently, or at least before communication resumes. Otherwise, the sites will not be able to establish communication.

Switchover procedures differ depending on whether the Primary and Secondary volumes are connected or not. As a general rule:

- When the coupling is deactivated, it is acceptable to change the role on one side only, assuming that the other side will be changed as well before communication resumes.
- If the coupling is activated, but is either unsynchronized or nonoperational due to a link error, an administrator must either wait for the coupling to be synchronized, or deactivate the coupling.
- On the Secondary volume, an administrator can change the role even if coupling is active. It is assumed that the coupling will be deactivated on the Primary volume and the role switch will be performed there as well in parallel. If not, a configuration error occurs on the original Primary volume.

Changing the Secondary volume to the Primary

The role of the Secondary volume can be changed to Primary, using Hyper-Scale Manager or CLI. After this switchover, the following takes effect:

- The Secondary volume is now the Primary volume.
- The coupling has the status of unsynchronized.
- The coupling remains in Standby mode, meaning that the remote mirroring is deactivated. This ensures an orderly activation when the role of the other site is switched.

The new Primary volume starts to accept write commands from local hosts. Because coupling is not active, in the same way as any Primary volume, it maintains a log of which write operations should be sent to the Secondary volume when communication resumes.

Typically, after switching the Secondary to the Primary volume, an administrator also switches the Primary to the Secondary volume, at least before communication resumes. If both volumes are left with the same role, a configuration error occurs.

Changing the Primary volume to the Secondary

When coupling is inactive, the primary machine can switch roles. After such a switch, the Primary volume becomes the Secondary.

Before switching roles, the Primary volume is inactive. Hence, it is in the unsynchronized state, and it might contain data that has not been replicated. Such data will be lost. When the Primary volume becomes Secondary, this data must be discarded to match the data on the peer volume, which is now the new Primary volume. In this case, an event is created, summarizing the size of the lost data.

Upon reestablishing the connection, the recovery volume (current Secondary, which was the Primary) will update the remote volume (new Primary) with this uncommitted data list to update, and it is the responsibility of the new Primary volume to synchronize these lists to the local volume (new Secondary).

I/O operations in synchronous mirroring

I/O operations are performed on the Primary and Secondary volumes across various configuration options.

I/O on the Primary volume

Read All data is read from the primary (local) site regardless of whether the system is synchronized.

Write

- If the coupling is operational, data is written to both the Primary and Secondary volumes.
- If the coupling is non-operational, data is written to the Primary volume only, and the Primary is aware that the Secondary is currently not synchronized.

I/O on the Secondary volume

The LUN of a Secondary volume can be mapped to remote hosts. In this case, the Secondary volume will be accessible to those remote hosts as *Read-only*.

These mappings are then used by remote hosts for Primary-Secondary role switchover. When the Secondary volume becomes the Primary, hosts can write to it on the remote site. When the Primary volume becomes a Secondary volume, it becomes *Read-only* and can be updated only by data replicated from the new Primary volume.

Read Data can be read from the Secondary volume like from any other volume.

Write

In an attempt to write on the Secondary volume, the host will receive a volume read-only SCSI error.

Synchronization speed optimization

The storage system has two global parameters that limit the maximum rate used for initial synchronization and for synchronization after non-operational coupling.

These limits are used to prevent a situation where synchronization uses too much of the system or communication line resources, and hampers the host's I/O performance.

The values of these global parameters can be viewed by the user, but setting or changing them should be performed by an IBM technical support representative.

Dynamic rate adaptation

The storage system provides a mechanism for handling insufficient bandwidth and external connections whenever remote mirroring is used.

The mirroring process replicates data from one site to the other. To accomplish this, the process depends on the availability of bandwidth between the local and remote storage systems. The mirroring synchronization rate parameter determines the bandwidth that is required for a successful mirroring.

You can request that an IBM technical support representative manually modify this parameter. To define its value, the IBM technical support representative should take into account the availability of bandwidth for the mirroring process, where the storage system adjusts itself to the available bandwidth.

The storage system prevents I/O timeouts through continuously measuring the I/O latency. Excessive incoming I/Os are queued until they can be submitted. The mirroring rate dynamically adapts to the number of queued incoming I/Os, allowing for a smooth operation of the mirroring process.

Implications on volume and snapshot management

When using sync mirroring, the default behavior of volumes and snapshots changes in order to protect the mirroring operation, as follows:

- Renaming a volume changes the name of the last-consistent and most updated snapshots.
- Deleting all snapshots does not delete the last-consistent and most updated snapshots.
- Resizing a Primary volume automatically resizes its Secondary volume.
- A Primary volume cannot be resized when the link is down.
- Resizing, deleting, and formatting are not permitted on a Secondary volume.
- A Primary volume cannot be formatted. If a Primary volume must be formatted, an administrator must first deactivate the mirroring, delete the mirroring, format both the Secondary and Primary volumes, and then define the mirroring again.
- Secondary or Primary volumes cannot be the target of a copy operation.
- Locking and unlocking are not permitted on a Secondary volume.
- The last-consistent and most updated snapshots cannot be unlocked.
- Deleting is not permitted on a Primary volume.
- Restoring from a snapshot is not permitted on a Primary volume.
- Restoring from a snapshot is not permitted on a Secondary volume.
- A snapshot cannot be created with the same name as the last-consistent or most updated snapshot.

Coupling synchronization process

When a failure condition has been resolved, remote mirroring begins the process of synchronizing the coupling. This process updates the Secondary volume with all the changes that occurred while the coupling was not operational.

The following diagram shows the various coupling states, together with the actions that are performed in each state.

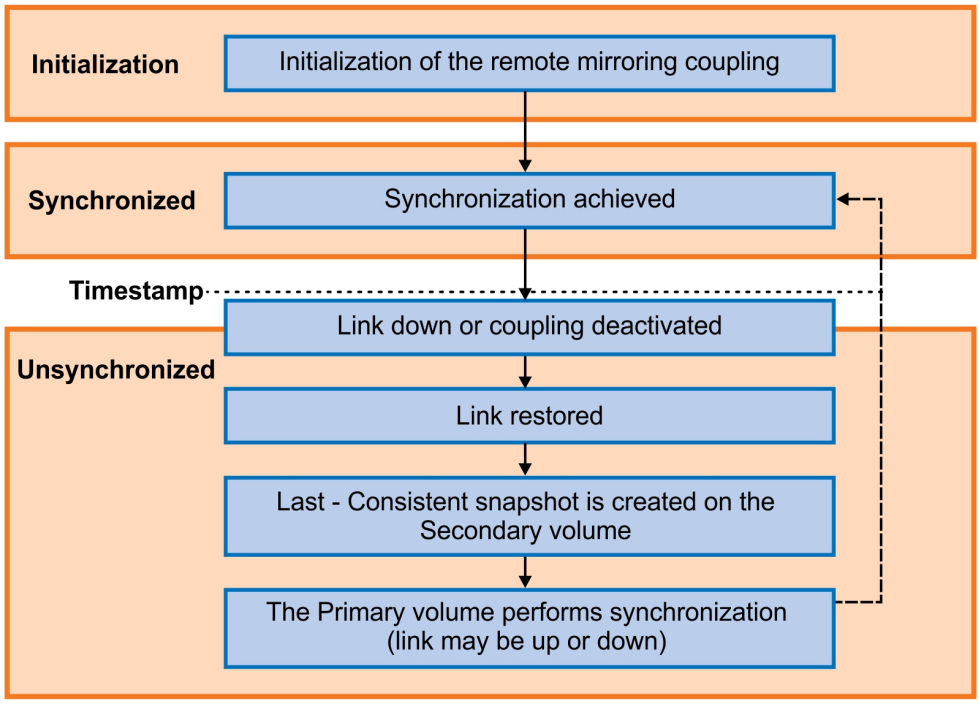


Figure 32. Coupling states and actions

The following list describes each coupling state:

Initialization
The Secondary volume has a *Synchronization* status of *Initialization*. During this state, data from the Primary volume is copied to the Secondary volume.

Synchronized
This is the working state of the coupling, where the data in the Secondary volume is consistent with the data in the Primary volume.

Timestamp
When a link is down, or when a coupling is deactivated, a timestamp needs to be taken. After the timestamp is taken, the state changes to *Timestamp*, and stays so until the link is restored, or the coupling is reactivated.

Unsynchronized

Remote mirroring is recovering from a communications failure or deactivation. The Primary and Secondary volumes are being synchronized.

Coupling recovery

When remote mirroring recovers from a non-operational coupling, the following actions take place:

- If the Secondary volume is in the *Synchronized* state, a last-consistent snapshot of the Secondary volume is created and named with the string `secondary-volume-time-date-consistent-state`.
- The Primary volume updates the Secondary volume until it reaches the *Synchronized* state.
- When all couplings that mirror volumes between the same pair of systems are synchronized, the Primary volume deletes the special snapshot.

Uncommitted data

For best-effort coupling, when the coupling is in *Unsynchronized* state, the system must track which data in the Primary volume has been changed, so that these changes can be committed to the Secondary when the coupling becomes operational again.

The parts of the Primary volume that must be committed to the Secondary volume and must be marked are called *uncommitted data*.

Constraints and limitations

The following constraints and limitations apply to the synchronization process:

- The size, part, or time-to-synchronize are relevant only if the synchronization status is *Unsynchronized*.
- The last-secondary time stamp is only relevant if the coupling is *Unsynchronized*.

Synchronous mirroring of consistency groups

Mirroring can be applied to whole consistency groups.

The following restrictions apply:

- All volumes in a consistency group have the same role, either Primary, or Secondary
- All mirrors in a consistency group are between the same two systems

Chapter 12. Asynchronous remote mirroring

Asynchronous mirroring enables high availability of critical data by asynchronously replicating data updates from a primary storage peer to a remote, secondary peer.

The relative merits of asynchronous and synchronous mirroring are best illustrated by examining them in the context of two critical objectives:

- Responsiveness of the storage system
- Currency of mirrored data

With synchronous mirroring, host writes are acknowledged by the storage system only after being recorded on both peers in a mirroring relationship. This yields high currency of mirrored data (both mirroring peers have the same data), yet results in less than optimal system responsiveness because the local peer cannot acknowledge the host write until the remote peer acknowledges it. This type of process incurs latency that increases as the distance between peers increases, but both peers are synchronized (first image below).

Asynchronous mirroring (second image below) is advantageous in situations that warrant replication between distant sites because it eliminates the latency inherent to synchronous mirroring, and might lower implementation costs. Careful planning of asynchronous mirroring can minimize the currency gap between mirroring peers, and can help realize better data availability and cost savings.

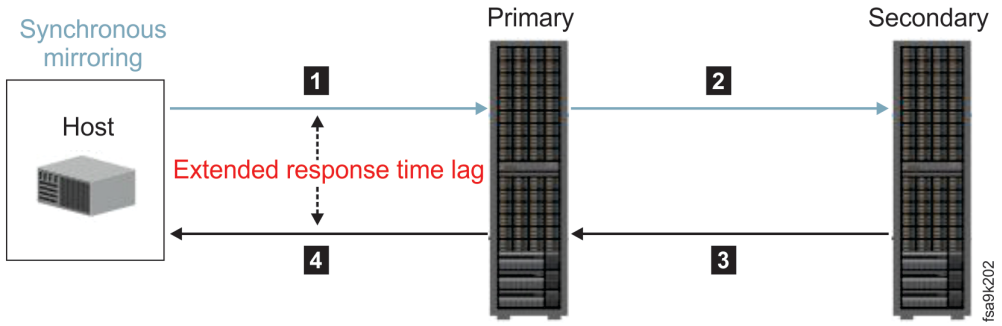


Figure 33. Synchronous remote mirroring concept

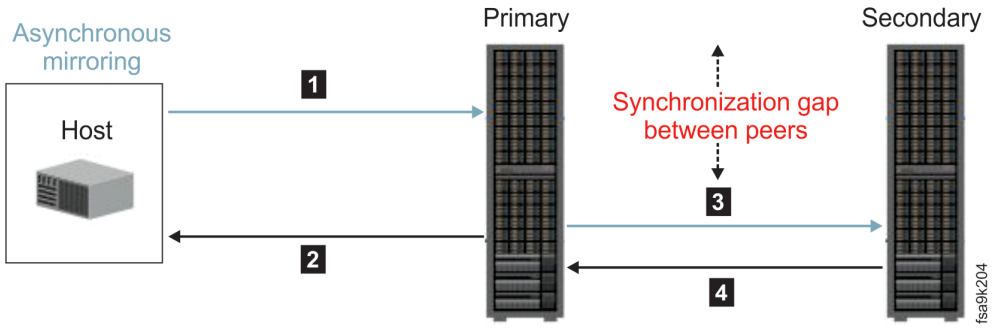


Figure 34. Asynchronous mirroring - no extended response time lag

Note: Synchronous mirroring is covered in Chapter 11, "Synchronous remote mirroring," on page 55.

For the information on the cross-system use of asynchronous remote mirroring, see the feature availability matrix.

Asynchronous mirroring highlights

The following are highlights of the asynchronous mirroring capability.

Advanced snapshot-based technology
Asynchronous mirroring is based on IBM snapshot technology, which streamlines implementation while minimizing impact on general system performance. The technology leverages functionality that supports mirroring of complete systems, translating to hundreds of mirrors. For a detailed description, see "Snapshot-based technology in asynchronous mirroring" on page 69.

Cross-generation asynchronous replication between A9000 systems and XIV Gen3
Cross-generation asynchronous replication between your A9000 systems and XIV Gen3 systems is simplified by the IBM Hyper-Scale Manager user-interface, allowing you to leverage your XIV Gen3 investment, and lower the cost of data protection and disaster recovery.

Mirroring of consistency groups
The storage system supports definition of mirrored consistency groups. In a consistency group, the mirrored image of the volumes at the remote site is kept consistent to the same point in time for all the volumes in the group. This is highly advantageous to enterprises, because it facilitates easy management of replication for all volumes that belong to a single consistency group. This also enables streamlined restoration of consistent volume groups from a remote site upon unavailability of the primary site.

Automatic and manual replication
Asynchronous mirrors can be assigned a user-configurable schedule for automatic, interval-based replication of changes, or can be configured to replicate changes upon issuance of a manual (or scripted) user command. Automatic replication allows you to establish crash-consistent replicas, whereas manual replication allows you to establish application-consistent replicas, if required. You can combine both approaches, because you can define mirrors with a scheduled replication and issue manual replication jobs for these mirrors as needed.

Multiple RPOs (Recovery Point Objectives) and multiple schedules
Asynchronous mirroring enables each mirror to be specified a different RPO, rather than forcing a single RPO for all mirrors. This can be used to prioritize replication of some mirrors over others, potentially making it easier to accommodate application RPO requirements, as well as bandwidth constraints.

Flexible and independent mirroring intervals
Asynchronous mirroring supports schedules with intervals ranging between 20 seconds and 12 hours. Moreover, intervals are independent from the mirroring RPO. This enhances the ability to fine tune replication to accommodate bandwidth constraints and different RPOs.

Flexible pool management
Asynchronous mirroring enables the mirroring of volumes and consistency groups that are stored in thin provisioned pools. This applies to both mirroring peers.

Bi-directional mirroring
A storage system can host multiple mirror sources and targets concurrently, supporting over a thousand mirrors per system. Furthermore, any given storage system can have mirroring relationships with several other storage systems. This enables enormous flexibility when setting mirroring configurations.

The number of systems with which the storage system can have mirroring relationships is specified outside, in the Data Sheet.

Concurrent synchronous and asynchronous mirroring
The storage system can concurrently run synchronous and asynchronous mirrors.

Easy transition between peer roles
Mirror peers can be easily changed between Primary and Secondary.

Easy transition from independent volume mirrors into consistency group mirror
The asynchronous mirroring allows for easy configuration of consistency group mirrors, easy addition of mirrored volumes into a mirrored consistency group, and easy removal of a volume from a mirrored consistency group while preserving mirroring for such volume.

Control over synchronization rates per target
The asynchronous mirroring implementation enables administrators to configure different system mirroring rates with each target system.

Comprehensive monitoring and events
Storage systems generate events and monitor critical asynchronous mirroring-related processes to produce important data that can be used to assess the mirroring performance.

Easy automation via scripts
All asynchronous mirroring commands can be automated through scripts.

Snapshot-based technology in asynchronous mirroring

A snapshot-based technology for asynchronous mirroring facilitates concurrent mirrors with different recovery objectives.

With asynchronous mirroring, write order on the Primary volume is not preserved on the Secondary volume. As a result, a snapshot taken of the Secondary volume at any moment is most likely inconsistent and therefore not valid. To ensure high

availability of data in the event of a failure or unavailability of the Primary volume, it is imperative to maintain a consistent replica of the Primary volume that can ensure service continuity.

This is achieved through storage system snapshots. Asynchronous mirroring employs them to record the state of the Primary volume, and calculates the difference between successive snapshots to determine the data that needs be copied from the Primary to the Secondary volume as part of a corresponding replication process. Upon completion of the replication process, a snapshot is taken of the Secondary volume and reflects a valid replica of the Primary volume.

Below are select technological properties that explain how the snapshot-based technology helps realize effective asynchronous mirroring:

- The storage system supports a practically unlimited number of snapshots, which facilitates mirroring of complete systems with practically no limitation on the number of mirrored volumes supported
- Asynchronous mirroring implements memory optimization techniques that further maximize the performance attainable by minimizing disk access.

Disaster recovery scenarios in asynchronous mirroring

A disaster is a situation where one of the sites (either Primary or Secondary) fails, or the communication between the Primary site and the Secondary site is lost.

Asynchronous mirroring attains synchronization between the Primary and Secondary peers through a recurring data replication process called a *Sync Job*. Running at user-configurable schedules, the *Sync Job* takes the most recent snapshot of the Primary volume and compares this snapshot with the last replicated snapshot on the Secondary volume. The *Sync Job* then synchronizes the Primary volume data corresponding to these differences with the Secondary volume. At the completion of a sync job, a new last replicated snapshot is created on both the Secondary and Primary volumes.

Disaster recovery scenarios handle cases in which one of the snapshots mentioned above becomes unavailable. These cases are:

Unplanned service disruption

1 Failover

Unplanned service disruption starts with a failover to the Secondary volume.

The Secondary is promoted and becomes the new Primary, serving host requests

2 Recovery

Next, whenever the Primary volume and the link are restored, the replication is set from the promoted Secondary (the new Primary) onto the demoted Primary (the new Secondary).

Alternatively: No recovery

If recovery is not possible, a new mirroring is established on the Secondary volume. The original mirroring is deleted and a new mirroring relationship is defined.

3 Failback

Following the recovery, the original mirroring configuration is reestablished. The Primary volume maintains its role and replicates to the Secondary volume.

Planned service disruption

1 Planned role switch

Planned service disruption starts with a coordinated demotion of the Primary to the Secondary, while the Secondary is promoted to become the new Primary. The promoted Secondary volume serves host requests, and replicates to the demoted Primary. On the host side, the host is disconnected from the demoted Primary and connected to the new Primary.

2 Recovery

Next, whenever the Primary volume and the link are restored, the replication is set from the promoted Secondary volume (the new Primary) onto the demoted Primary (the new Secondary volume).

2 Failback

Following the recovery, the original mirroring configuration is reestablished. The Primary volume maintains its role and replicates to the Secondary volume.

Testing

There are two ways to test the Secondary volume replica:

- Create a snapshot of an LRS snapshot on the Secondary volume. Then map a host to it and verify the data.
- Disconnect the host from the Primary volume, switch roles, and connect the host to the Secondary volume. This is a more realistic, but also a more disruptive test.

Note: Please contact IBM Support in case of disaster or for any testing of disaster recovery, in order to get clear guidelines and to secure a successful test.

Chapter 13. High availability with HyperSwap

HyperSwap delivers highly-available, non-disruptive storage service, through partial or complete system failures and disasters, in the same data center and between metro-distant data centers.

HyperSwap high availability is based on active-active pairing of storage systems per volume or per consistency group. Each volume or consistency group pair uses synchronous replication to keep both systems updated at all times.

When certain conditions apply, an automatic and completely transparent failover is performed, so that the applications experience no downtime. As soon as the actual failure is recovered, the pair is automatically resynchronized.

As in other high availability solutions, HyperSwap requires a quorum witness component, to avoid split-brain situations. HyperSwap Quorum Witness is constantly monitoring the status of the related storage systems, and, if necessary, acts as a tiebreaker for conflict resolution.

The HyperSwap solution relies on Asymmetrical Logical Unit Access (ALUA) support to inform the host about the optimized paths to the storage system, and minimize I/O latency.

FlashSystem A9000 and FlashSystem A9000R HyperSwap capability does not require additional special hardware or software, and does not require any additional licensing.

For the information on the cross-system use of HyperSwap, see the feature availability matrix.

Design guidelines

To properly configure and implement high availability with IBM HyperSwap, it is important to understand its principles and key components.

This section provides general information about the IBM HyperSwap main components, configuration topology, and supported networking. It also describes the HyperSwap volume, and explains in which situations an automatic failover is expected.

Configuration

This sub-section provides general information about the IBM HyperSwap solution components, the concept of a failure domain, the HyperSwap volume, the Quorum Witness, and configuration requirements.

A minimal HyperSwap solution consists of:

- Two IBM FlashSystem A9000 or A9000R storage systems, interconnected for synchronous replication via Fibre Channel
- HyperSwap-protected hosts, each connected to both systems via iSCSI or Fibre Channel
- Quorum Witness software, installed on a VM or a physical host, with TCP-IP connectivity to both systems.

The paired systems maintain one or more HyperSwap relationships between them. Each relationship facilitates one HyperSwap volume (see "HyperSwap volumes and consistency groups" below).

Larger HyperSwap solutions are possible, since every IBM FlashSystem A9000/R system can have HyperSwap relationships with multiple other systems.

HyperSwap volumes and consistency groups

A HyperSwap volume is implemented as a pair of two volumes with identical SCSI attributes, one on each system.

These volumes should be always kept synchronized. From the host perspective, the two volumes are a single volume, and I/O – both reads and writes - can be served from either system, depending on the path used by the host. In other words, a HyperSwap solution allows a host to have active-active access to the same data on two systems.

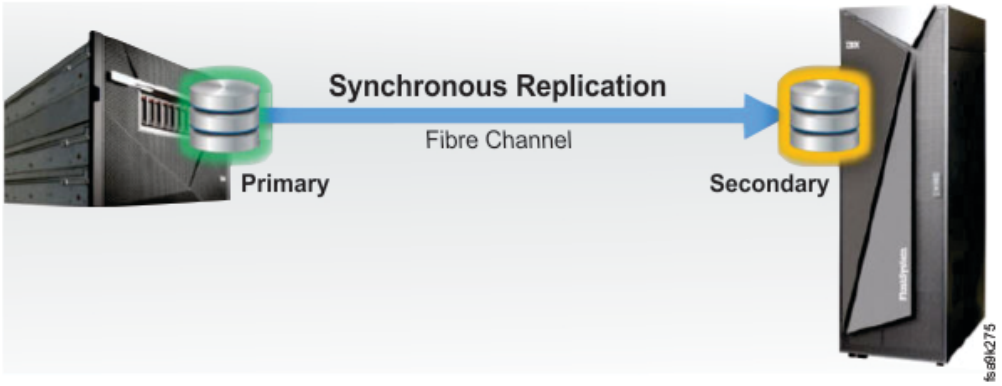


Figure 35. HyperSwap volume replication

- To be identical, a pair of volumes that constitute a HyperSwap volume have identical SCSI identity and I/O related attributes - size, locks and reservations. Each storage system maps the paired volumes to the host separately, but the host perceives them as a single volume. This makes transitions, such as automatic failover and manual failback, transparent both to the hosts and the applications running on them.
- To be synchronized, the peer-systems are interconnected for synchronous replication, and a HyperSwap relationship is established between the peer-volumes. One of these volumes is initially designated as Primary, and the other is designated as Secondary. As opposed to mirroring, the replication between the Primary and Secondary volumes is bi-directional, to allow read and write I/O to be served on either volume. The purpose of the Primary/Secondary designation is to optimize latency: the Primary volume should be co-located with the hosts that generate most of the I/O.
- The actual roles – Primary or Secondary - performed by the volumes at any given moment may differ from their original role designations, as a result of a manual role change or an automatic failover.

Multiple HyperSwap volumes can exist between any HyperSwap-paired systems. Moreover, HyperSwap volumes can be grouped into A9000/A9000R consistency groups. In such a case, HyperSwap actions will be applied to the entire consistency group, that is, to all the HyperSwap volumes in it.

The following restrictions apply to HyperSwap consistency groups:

- All the volumes in a HyperSwap consistency group must be HyperSwap volumes
- All the volumes in a HyperSwap consistency group must have the same target.

Important: When using IBM Hyper-Scale Mobility on Storage Area Network (SAN) Boot Volumes, the Proxy targets need to be replaced with the Owner targets on the adapter BIOS. In some cases, this can be done online from the operating system, by using vendor's online tools for x86 servers (for example, Qlogic QConvergeConsole), or by using specific commands to edit the boot paths order for Unix servers. In other cases, manual changes of the adapter BIOS are required.

Quorum Witness

Every clustering solution requires a Quorum Witness component that can be consulted by the cluster members at any time to avoid split-brain situations.

In the HyperSwap solution, this function is performed by the IBM Spectrum Accelerate Family HyperSwap Quorum Witness software application. It allows FlashSystem A9000 & A9000R arrays to determine in any conditions and at any time, for each HyperSwap volume, which single array should exclusively own the Primary volume instance.

Connectivity between a Quorum Witness and storage systems is established via TCP-IP, as shown in the Topologies section.

When the Quorum Witness is down for any reason, there is no impact on HyperSwap active-active data access, and various failure scenarios will still be accommodated without disruption. However, during that period, automatic failover cannot be applied, since the A9000 & A9000R systems have no other way to ensure that a failover will not result in a split-brain situation. Therefore when the Quorum Witness is down, the risk of downtime is elevated.

To minimize Quorum Witness downtime, it should be continuously monitored, and any issues should be addressed immediately. To further minimize the risk of Quorum Witness downtime, it can be deployed as a highly-available VM on a VMware vSphere cluster, using VMware High Availability.

Failure domain

A failure domain encompasses all the elements potentially affected by a single failure.

For example, an earthquake or power-grid failure can potentially take down a whole datacenter. The datacenter is therefore a failure domain, with regards to earthquake and power-grid failures. To protect against such failures, the infrastructure can be divided between two geographically-separate data centers that are not using the same power grid. Each data center is considered a failure domain, and if one of them fails, the other can replace it. Failure domains can also be defined within a single datacenter, depending on the kind of failure they protect against. For example, to protect against overheating, the datacenter may have multiple cooling systems. The area that is protected by each cooling system will be a failure domain, from cooling perspective.

Therefore, for every two systems that are HyperSwap-paired, the best practice is to separate and spread the two systems and the Quorum Witness across three failure-domains to prevent the situation where more than one of these elements is

down. The highest availability level is obtained when these three failure domains are 3 geographically separated sites, with separate power and network resources.

Topologies

This section surveys various configurations of hosts, storage systems and Quorum Witnesses for HyperSwap volumes.

Volume level perspective

This section focuses on a typical configuration of hosts, storage systems and a Quorum Witness for one HyperSwap volume.

Typically, the host and the storage system initially designated for Primary volume are located at the same site, and the Quorum Witness is deployed at a separate third site.

In the diagram below, the choice of FlashSystem A9000R on one side, and FlashSystemA9000 on the other side is not necessarily typical, it is only used to emphasize that it is a valid configuration.

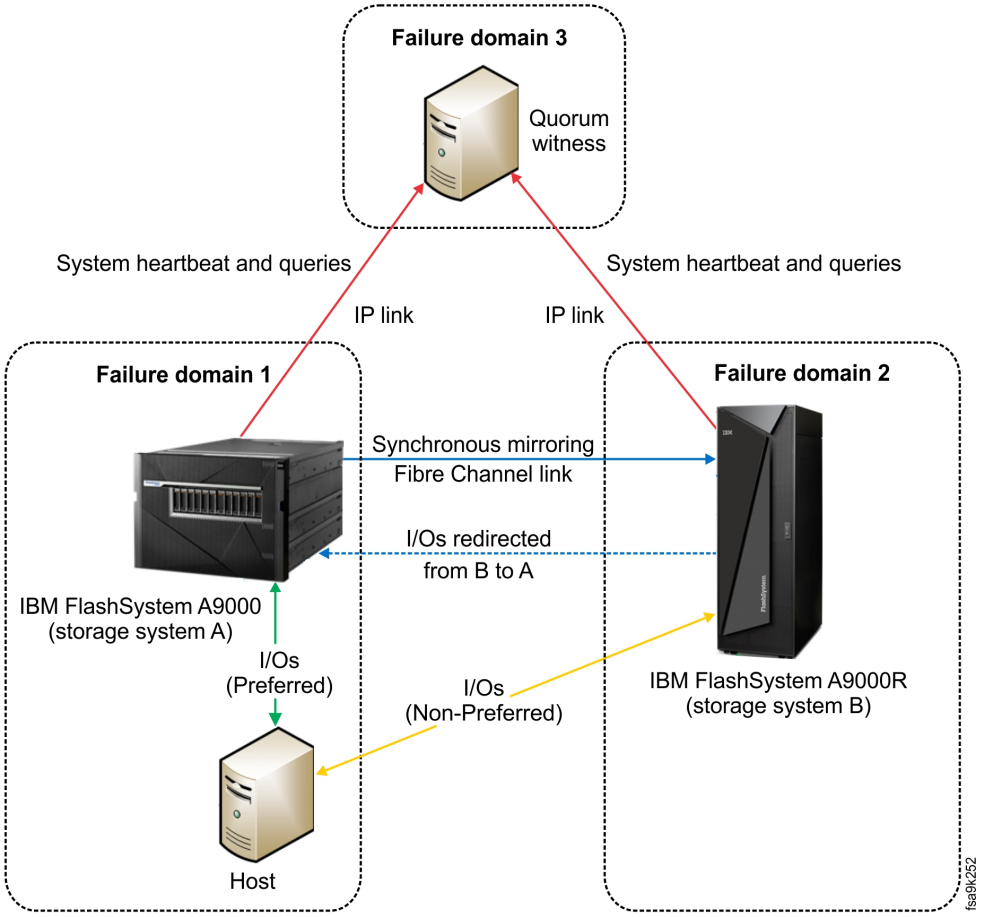


Figure 36. Typical HyperSwap high availability configuration

Host-to-storage-system paths, hereafter referred to as port groups, are optimized using Asymmetric Logical Unit Access (ALUA) support from the multipath driver.

By assigning proper ALUA states (Preferred or Non-Preferred), the storage system directs the host multipath driver which paths are preferred, to minimize I/O latency:

- Port groups to the system that currently owns the Primary volume are automatically marked as Active/Preferred
- Port groups to the system that currently owns the Secondary volume are automatically marked as Active/Non-Preferred.

As a result, Active/Preferred port groups receive the bulk of the I/O traffic. Any remaining I/O traffic is directed to the Active/Non-Preferred port groups, and writes are then forwarded to the Primary volume.

When the HyperSwap volume is activated, the Secondary volume is not synchronized, and read requests are redirected to the Primary volume until synchronization has completed. When the volumes are synchronized, the system that owns the Secondary volume serves read requests locally.

If the system that owns the Secondary volume is unable to perform I/O, the Secondary volume port group state changes to Unavailable. Such a change would typically be a result of a connectivity failure. The Primary volume remains active, therefore no automatic failover is needed. As soon as connectivity is restored, the volumes will be re-synchronized automatically.

If the system that owns the Primary volume is unable to perform I/O, the Primary volume port group state changes to Unavailable. As soon as the system that owns the Secondary volume receives the corresponding notification from the Quorum Witness, it performs a transparent failover, by which the Secondary volume assumes the Primary role. When the system that was originally designated for the Primary volume is restored, recovery must be performed manually via CLI or via IBM Hyper-Scale Manager, which involves switching the roles of the peer volumes and re-activating the HyperSwap relationship between them.

System level perspective

Since every system can contain a mix of multiple Primary and Secondary volumes and have HyperSwap relationships with multiple systems, it is interesting to consider topologies at system level, with multiple HyperSwap volumes.

Here are a few examples, where green denotes Primary volumes, and yellow denotes Secondary volumes.

In a conventional Disaster Recovery topology (that is, where one system is located in a site that is designated by the storage operations for disaster recovery) one storage system owns the volumes designated as Primary, and the other system owns the volumes designated as Secondary:

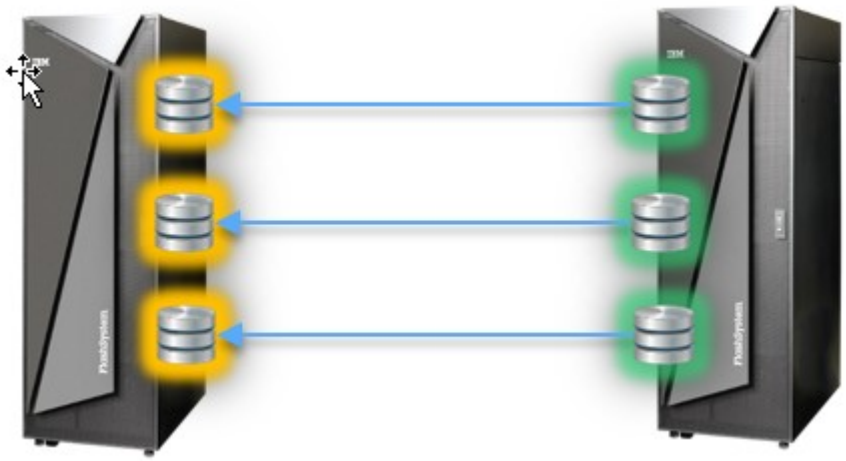


Figure 37. HyperSwap topology: conventional

In a star-shaped Disaster Recovery topology, where one of the sites is designated by the storage operations as a Disaster Recovery site, a single storage system in the Disaster Recovery site is dedicated to simultaneously serving multiple other systems:

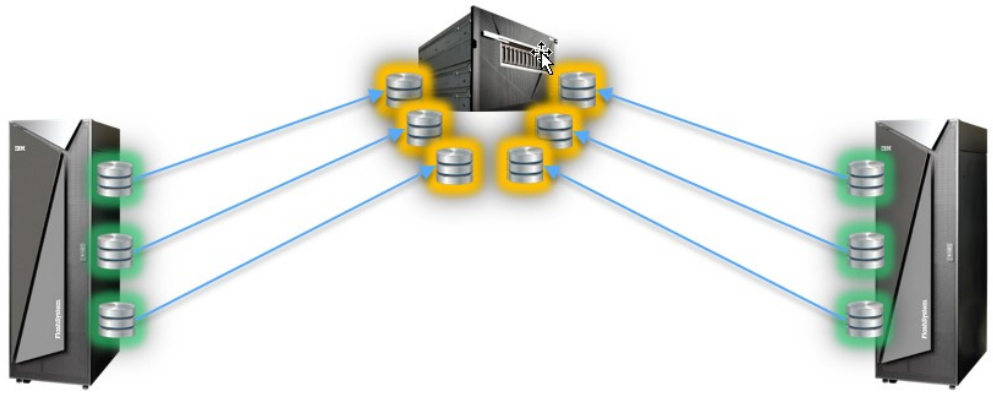


Figure 38. HyperSwap topology: dedicated

In a symmetrical system topology, both systems have volumes designated as Primary and Secondary, depending on the preferred location of the application:

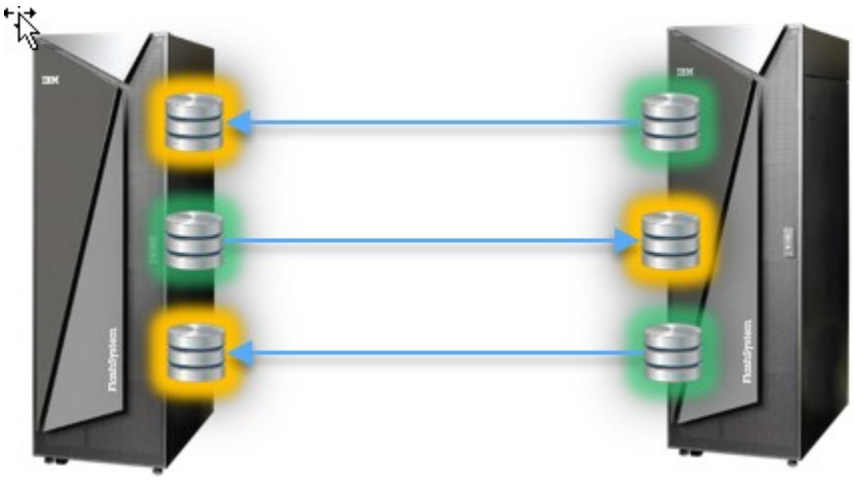


Figure 39. HyperSwap topology: symmetrical

Other topologies

Since every system can have HyperSwap relationships with up to ten other systems, and serve a mix of Primary and Secondary volumes, other topologies are possible as well.

Two types of configuration of HyperSwap topology are possible based on how hosts are connected to storage systems.

In a uniform configuration, each host can access both the Primary and Secondary volumes. The uniform configuration is the best practice to protect a host from data access problems:

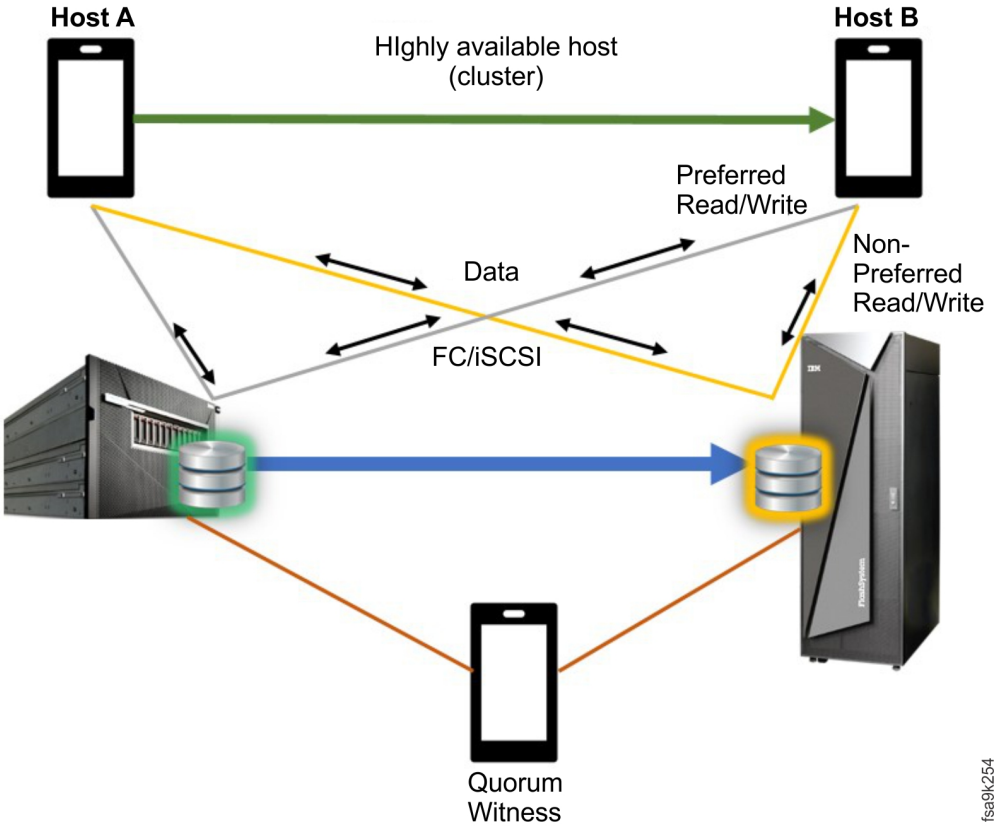


Figure 40. HyperSwap configuration types: Uniform host connectivity

In a non-uniform configuration, the storage high availability relies on the server detection and failover of the application to the server with access to active storage. A non-uniform configuration can be used when the host is part of a cluster, can fail over to another host in the cluster, and the other host is connected to the peer system. This configuration is less costly from network perspective. However, it relies on a complex failover orchestration between host-clusters and the FlashSystem A9000 or FlashSystem A9000R systems, which is not needed in a uniform configuration.

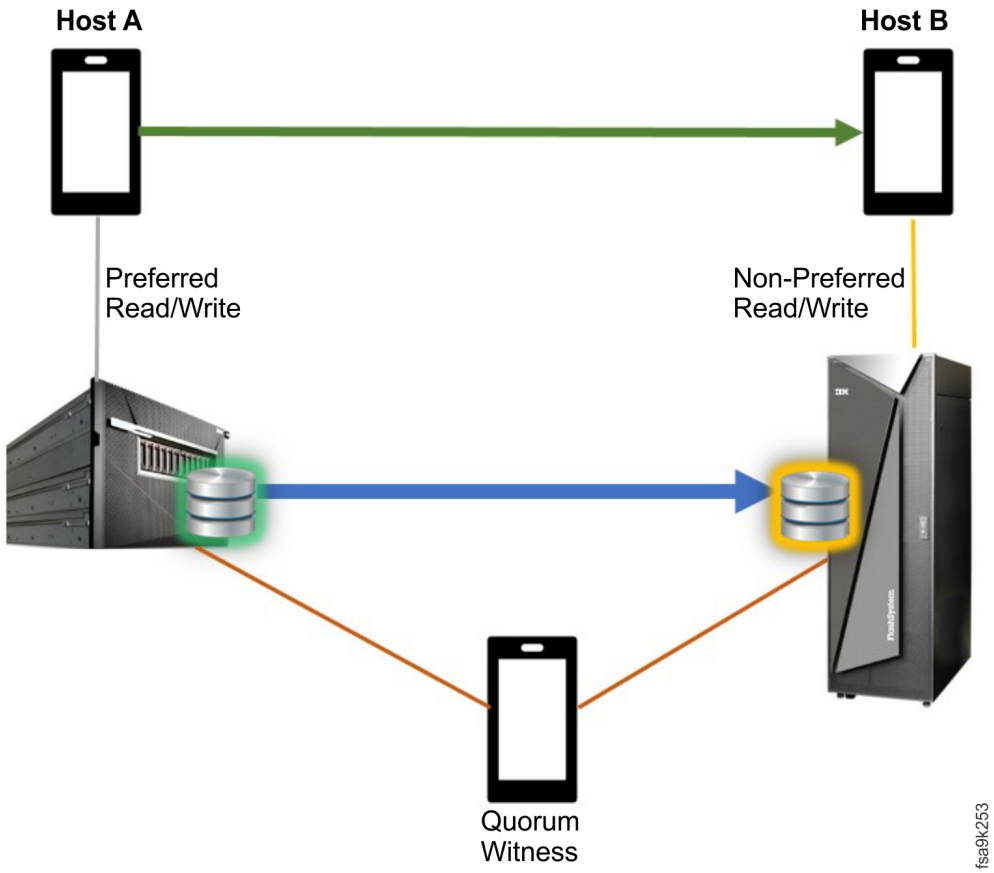


Figure 41. HyperSwap configuration types: Non-uniform host connectivity

Automatic failover scenarios

IBM FlashSystem A9000 and IBM FlashSystem A9000R HyperSwap solution is designed to maintain non-disruptive host data access.

The HyperSwap decision-making process guarantees non-disruptive data access, while avoiding unnecessary failover.

To achieve this, the HyperSwap logic considers two aspects:

- System and Quorum Witness failure identification
- Assignment of Primary and Secondary roles to HyperSwap volumes.

The Quorum Witness allows the peer systems to avoid making erroneous or even conflicting decisions regarding the role of each volume in a HyperSwap volume pair. For instance, if connectivity between them is disrupted, the system that owns the Secondary volume may assume that its peer system has failed, and attempt to initiate an automatic failover in order to become the owner of the Primary volume. If this attempt succeeds while the system that owns the Primary volume is still serving I/O, both systems will claim that they own the Primary volume. The Quorum Witness protects against such split-brain scenarios. If the Primary volume owner system is still communicating with the Quorum Witness, the peer system

will be aware of it. Instead of attempting a failover, it will render its own port groups *Unavailable*. Replication will stop and resume only after the connectivity problem is resolved.

Operating a HyperSwap-based High Availability solution

The following topics describe how to operate a HyperSwap-based High Availability solution.

Establishing a HyperSwap relationship

A HyperSwap relationship is similar in many ways to a mirroring relationship, but it has extra attributes and states, that are related to the HyperSwap volume and high availability logic.

Configuring a HyperSwap environment

Before provisioning the first HyperSwap volume, a HyperSwap environment must be configured.

Configuration of a HyperSwap environment begins with a Quorum Witness deployment, as described in the *Quorum Witness User Guide*. In IBM Hyper-Scale Manager, define the same Quorum Witness on all the participating peer systems using one simple configuration:

Figure 42. Quorum Witness information

When connectivity with the Quorum Witness is established, the system registers itself with the Quorum Witness, and the Quorum Witness is considered active by that system. Each peer system continuously verifies that it has connectivity with the target system and the Quorum Witness, and that the target system has connectivity with the same Quorum Witness.

Creating a new HyperSwap relationship

As a prerequisite for creating a HyperSwap relationship, the peer systems verify that:

- They are both registered on the same Quorum Witness

- The remote system is identified via the Quorum Witness as *healthy*
- The overall connectivity state is suitable for High Availability

If the checks are completed successfully, define the Primary and Secondary volumes and Consistency Groups to create a new HyperSwap relationship. As explained in Configuration, the Secondary volume in a HyperSwap relationship adopts the identity of the Primary volume.

Creating a HyperSwap relationship from an existing synchronous mirror

A synchronous mirror relationship can be transformed into a HyperSwap relationship, if the following requirements are met:

- All the required configuration has completed successfully: the Quorum Witness is configured and activated, and the remote target configuration is updated
- The Secondary volume is unmapped
- The Quorum Witness is active and connected
- Both mirror volume names are identical.

  If a Consistency Group is converted into a HyperSwap Consistency Group, the names of all the volumes in the Secondary Consistency Group must be identical with the names of their peers in the Primary Consistency Group.

The transformation can be carried out only when the mirror is active. It does not disrupt the mirroring, and the data on the Secondary volume remains intact.

Activating a HyperSwap relationship

When a HyperSwap relationship is activated, the Quorum Witness configurations are verified. After the verification has completed successfully, the Primary volume starts synchronizing data to the Secondary volume. When the relationship is activated, the Secondary volume becomes available, and active-active access is enabled.

The standard mirroring initialization methods are applicable to HyperSwap relationships:

- Online: This method is the default one. It uses an inter-site link to replicate the Primary volume's initial state to the Secondary volume, starting once the mirror is first activated.
- Offline: Initialization of the Secondary volume is not done by replicating the Primary volume's initial image, but rather by Offline Initialization. In other words, it restores to the Secondary volume a mirror image that was backed up from the Primary volume. Once the relationship is activated, the contents of the volumes are compared, and only modified data is synchronized over the wire. This process is usually much faster than online initialization, especially if the link throughput is limited.

For more information about mirroring initialization, see an IBM Redpaper publication IBM XIV Storage System Multi-Site Mirroring.

During initialization, the HyperSwap relation status is *Initializing*.

HyperSwap relationship between Consistency Groups

You can establish a HyperSwap relationship between HyperSwap Consistency Groups.

A HyperSwap Consistency Group is a Consistency Group that has a HyperSwap relationship with a remote Consistency Group. To be eligible for the addition to a HyperSwap Consistency Group, the volume must conform to the following requirements:

- The volume is a HyperSwap volume
- The volume role (Primary or Secondary) matches the role of the Consistency Group
- The volume is connected to the same target system or peer system as the Consistency Group
- The HyperSwap relationship is synchronized

Any later change to the Consistency Group, whether automatic or manual, affects all the volumes it contains. In particular, an automatic failover is carried out only if all the volumes in the Consistency Group are ready for the automatic failover.

Monitoring a HyperSwap volume

The IBM Hyper-Scale Manager can display the current status of a HyperSwap volume at any time.

The HyperSwap volume status is displayed in the **Volume Availability** window. The status depends on:

- Peer systems connectivity
- Availability of the Quorum Witness
- Ability of both Primary and Secondary volumes to handle I/O (*active-active* configuration)
- Synchronization between the Primary and Secondary volumes.

In the figure below, all the prerequisites for automatic failover readiness are fulfilled, as indicated by green check-boxes. Consequently, the **Auto. Failover Capability** box is green, which means that this volume is ready to perform an automatic failover when necessary.

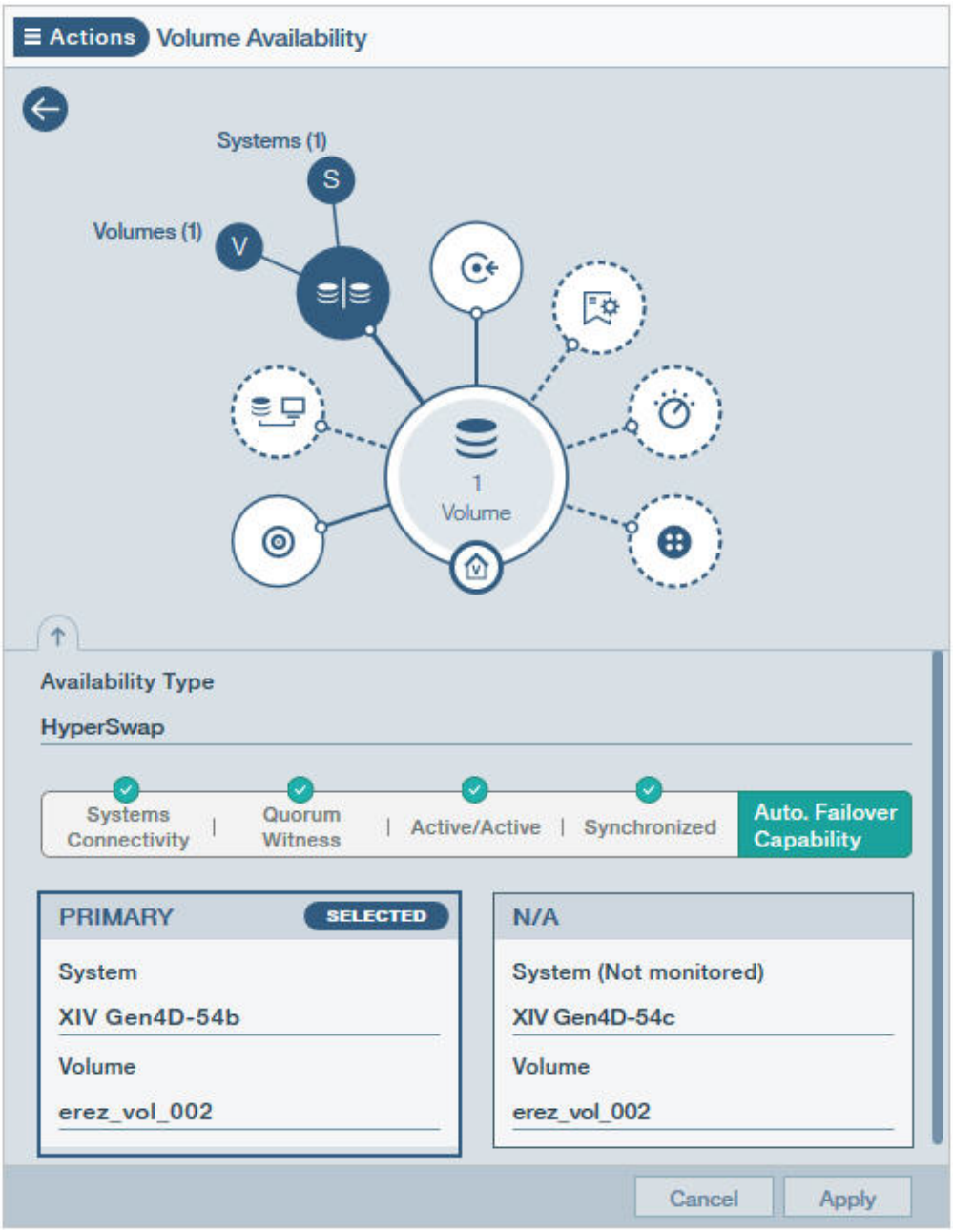


Figure 43. HyperSwap volume status

In addition, you can view the HyperSwap volume status in the **Group by Availability** column of the **Volumes** table.

Pointing at a volume marked as **H** in the **Group by Availability** column, displays a pop-up window with the volume's HyperSwap availability status:

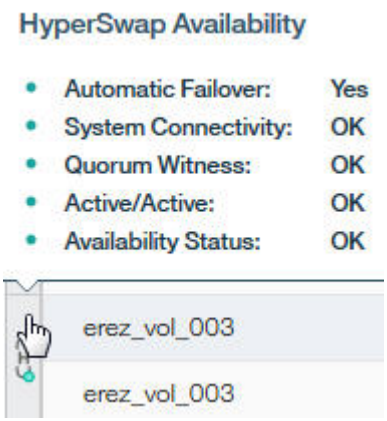


Figure 44. HyperSwap volume status in the Volumes table

Modifying a HyperSwap volume

Any change that affects the SCSI attributes of the HyperSwap volume is automatically carried out on both the Primary and Secondary volumes.

Volume attributes include WWN, name, size, locking status, and SCSI reservation. Therefore, having a healthy connectivity is mandatory for many HyperSwap volume related actions.

High Availability snapshots

HyperSwap supports the creation of identical single-point-in-time snapshots at the local and remote systems.

A snapshot is a logical volume reflecting the contents of a given source volume at a specific point-in-time. FlashSystem A9000 and A9000R storage systems use snapshots to identify inconsistencies that may arise between data updates. Snapshot taking and management are described in detail in “Volume function and lifecycle” on page 27.

In regular mirroring, snapshots are created on the local system, as described in “Synchronous mirroring operations” on page 56.

In addition to regular snapshots that can be applied to the Primary or Secondary volumes, HyperSwap supports the creation of identical single-point-in-time snapshots on the local and remote systems. The operation is performed on the Primary volume, but the resulting snapshots are created on both the local and remote systems. The content of these snapshots is identical, but unlike the Primary and Secondary volumes, that have identical SCSI attributes, the SCSI identities of those co-created High Availability snapshots are different. Therefore the snapshots are independent of each other, and any snapshot operations - such as snapshot deletion or unlocking - are performed on the individual snapshots. Moreover, if one of the snapshots is unlocked, and writes are applied to it, the content of the snapshots stops being synchronized and is no longer identical.

Failure detection and recovery scenarios

The IBM HyperSwap feature is designed to timely detect a failure and automatically choose the most appropriate course of action for fast and comprehensive recovery.

In the most sensitive scenario, upon detecting that the system that owns the Primary volume (hereafter referred to as System A) has failed, the system that owns the Secondary volume (hereafter referred to as System B) should trigger an automatic failover.

The specific failover scenarios and actions are described in the following sections.

System A - System B connectivity failure

If synchronous replication between Systems A and B is disrupted, System A remains solely responsible for serving I/O.

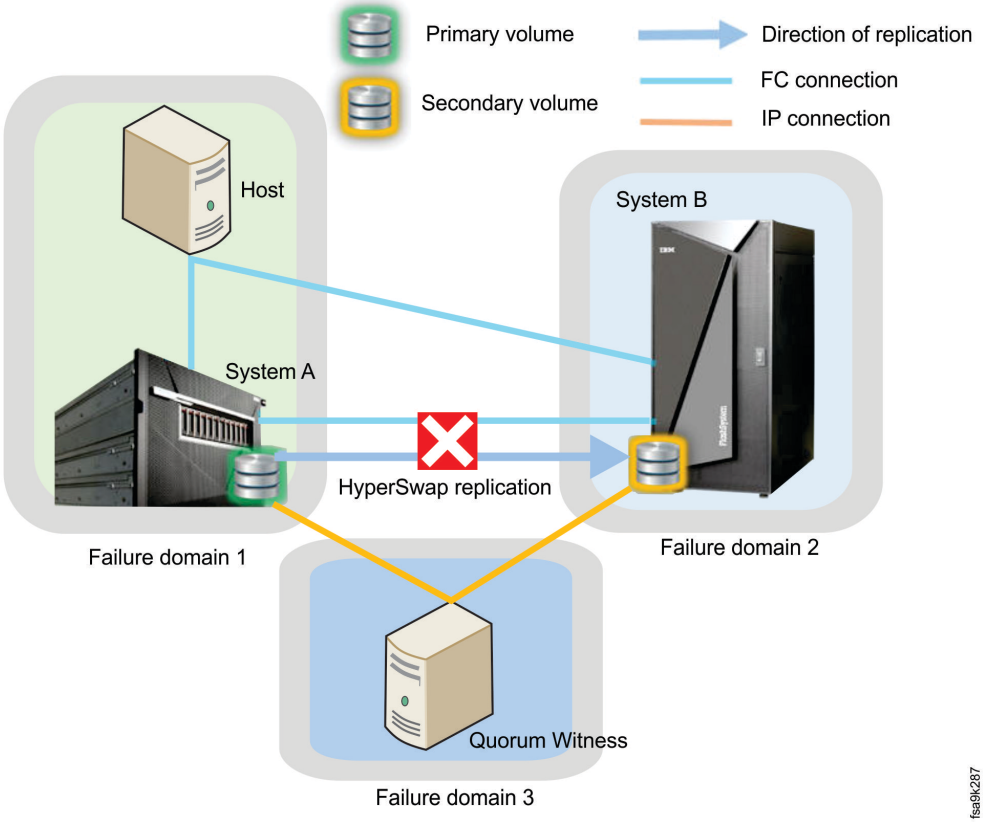


Figure 45. System A - System B connectivity failure

System A notifies the Quorum Witness about its taking sole responsibility for the relation. System B detects through the Quorum Witness that System A has taken sole responsibility for I/O, and stops serving I/O. It indicates to the host its inability to serve I/O by changing its port group state to *Unavailable*. At this point System B does not contain a synchronized image of the data and therefore is not ready for automatic failover.

As soon as connectivity is restored, the HyperSwap relationship automatically re-synchronizes the peer volumes. System B changes the state of its port groups to

active-preferred, and operates in redirect mode, meaning that it proxies all its I/O to System A. Once the relationship is synced, readiness for automatic failover is resumed, and the Secondary volume resumes serving read requests directly.

System A - Quorum Witness connectivity failure

If connectivity between System A and Quorum Witness is disrupted, System A will periodically attempt to recover connectivity. While Quorum Witness connectivity is not available, System B remains ready for automatic failover. Automatic failover will be triggered, if connectivity between the systems A and B fails as well.

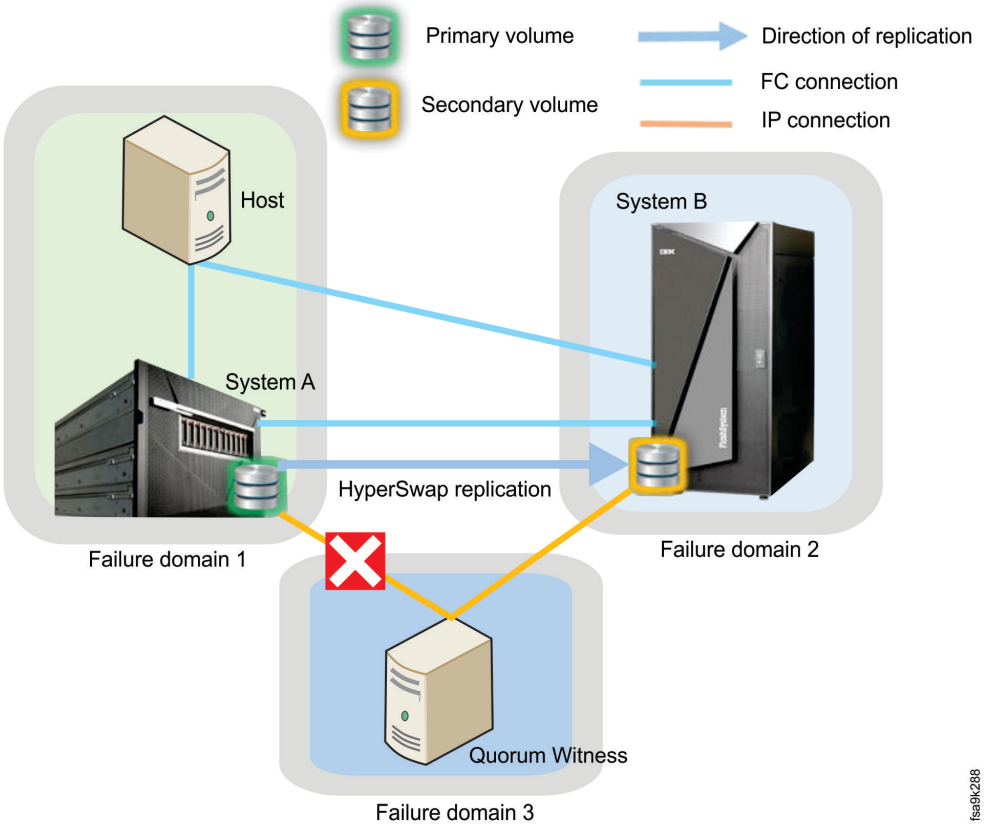


Figure 46. System A - Quorum Witness connectivity failure

If the problem is not resolved, an additional failure of the peer connectivity occurs, and automatic failover becomes impossible.

As soon as connectivity is recovered, System A will send a heartbeat to the Quorum Witness. System B continues performing as usual.

System A - System B and System A - Quorum Witness connectivity failure or System A failure

If connectivity between System A and the Quorum Witness is disrupted, and connectivity between System A and System B is down, System A stops serving I/O and assumes that System B performed an automatic failover.

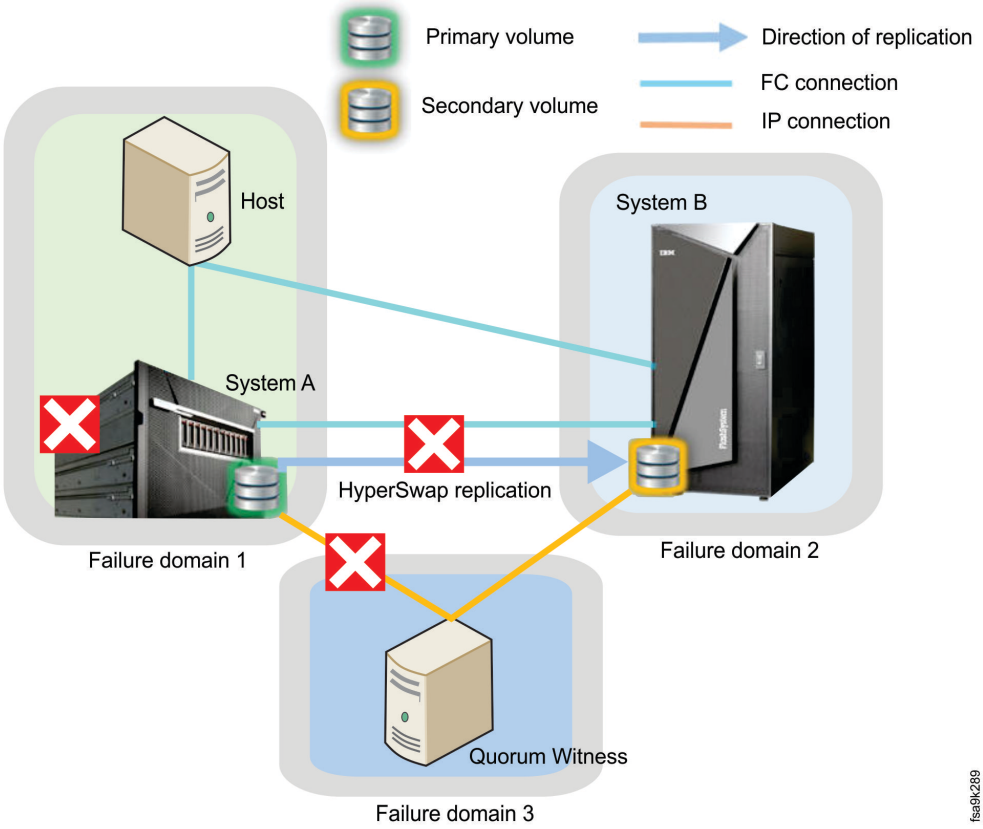


Figure 47. System A - System B and System A - Quorum Witness connectivity failure or System A failure

System B indeed triggers automatic failover by taking ownership of the Primary volume, that is by changing the role of the local volume from Secondary to Primary.

System A port group state changes to *Unavailable*. Automatic failover readiness is disabled, because only one copy of the data exists (on System B).

This scenario requires manual recovery. System A role must be changed to Secondary, and the HyperSwap relationship needs to be activated. At this point data synchronization will begin. After the data is synchronized, the roles can be switched in order to restore the original configuration.

System B - Quorum Witness connectivity failure

If connectivity between System B and Quorum Witness is disrupted, System A will continue performing as usual. System B will periodically attempt to recover connectivity with the Quorum Witness, but readiness for automatic failover will be disabled.

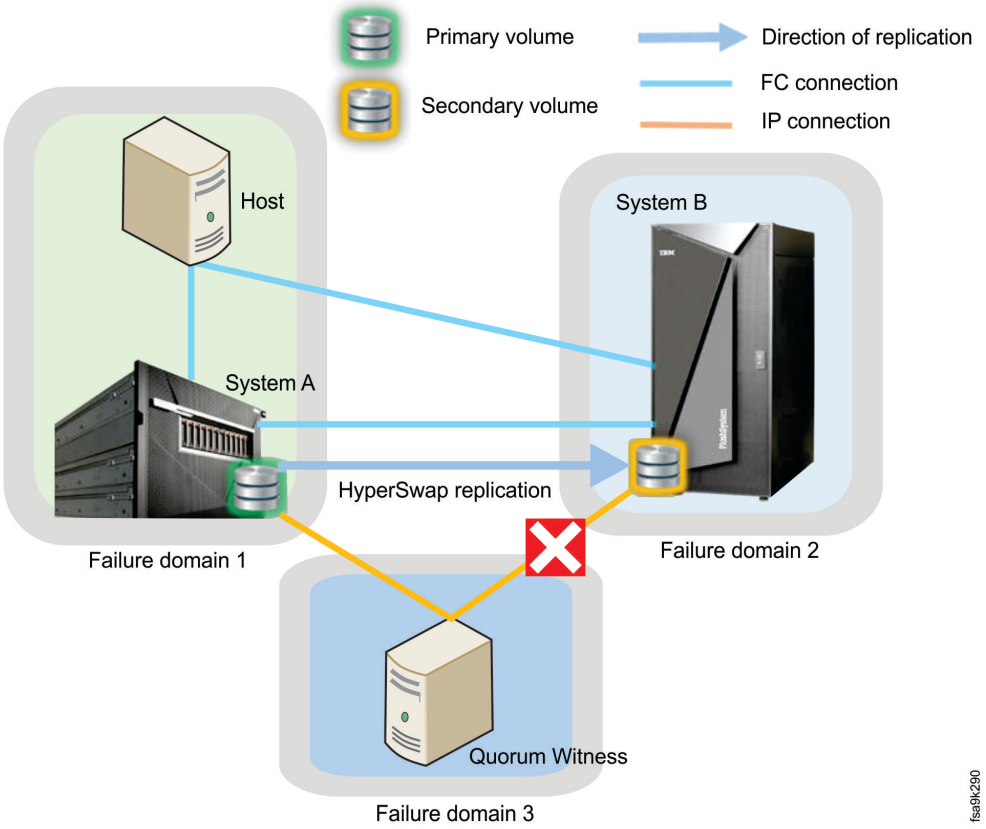


Figure 48. System B - Quorum Witness connectivity failure

When Quorum Witness connectivity is restored, readiness for automatic failover is immediately and automatically resumed.

System B - System A and System B - Quorum Witness connectivity failure or System B failure

If connectivity between System B and System A is disrupted, the replication is disrupted as well, and the Secondary volume goes out of sync. System A will then assume sole responsibility for serving I/O. Readiness for automatic failover will be disabled.

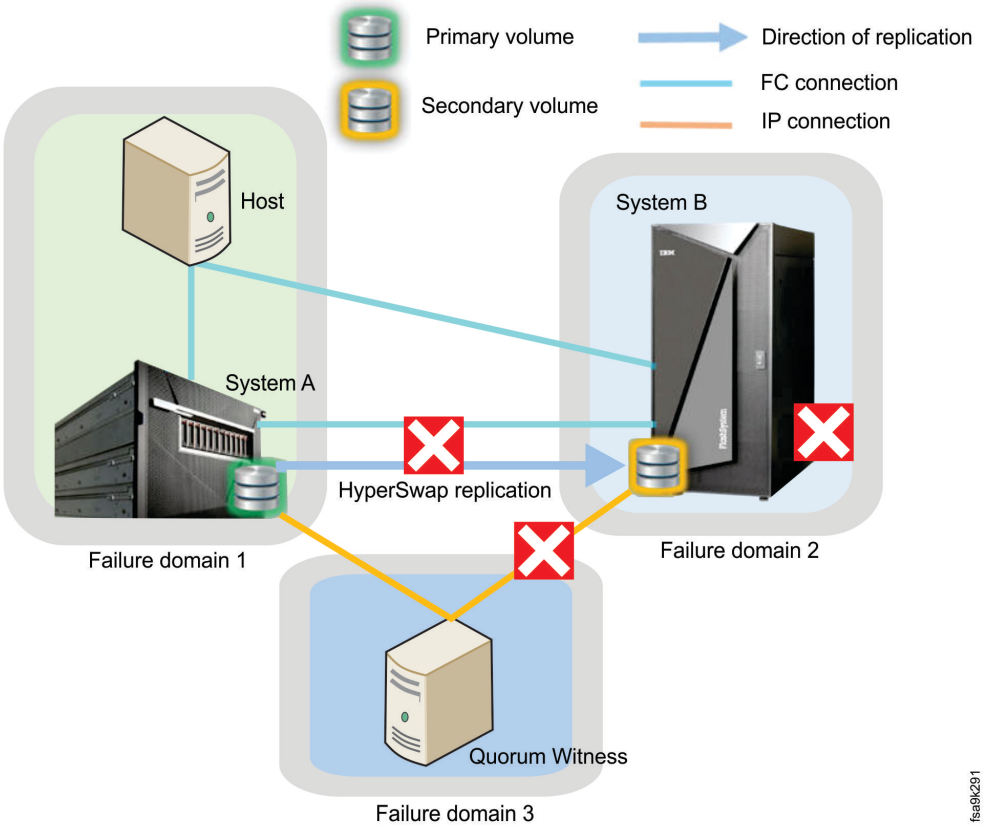


Figure 49. System B - System A and System B - Quorum Witness connectivity failure or System B failure

As soon as connectivity between System A and System B is restored, the volumes are automatically re-synchronized. System B changes the state of its port groups to *active/non-preferred*, and operates in redirect mode, meaning that it proxies any I/O to System A.

Once synchronization is complete and connectivity between System B and the Quorum Witness is restored, System B enables the HyperSwap relationship, and resumes handling read requests directly.

Automatic failover readiness is resumed once the data is synchronized and system B has connectivity to the Quorum Witness.

System A - Quorum Witness and System B - Quorum Witness connectivity failure

This scenario assumes that neither System A nor System B is connected to the Quorum Witness, while the System A - System B connectivity is in proper order. Automatic failover readiness will be disabled. System B, however, continues serving I/O, and data on both systems remains synchronized.

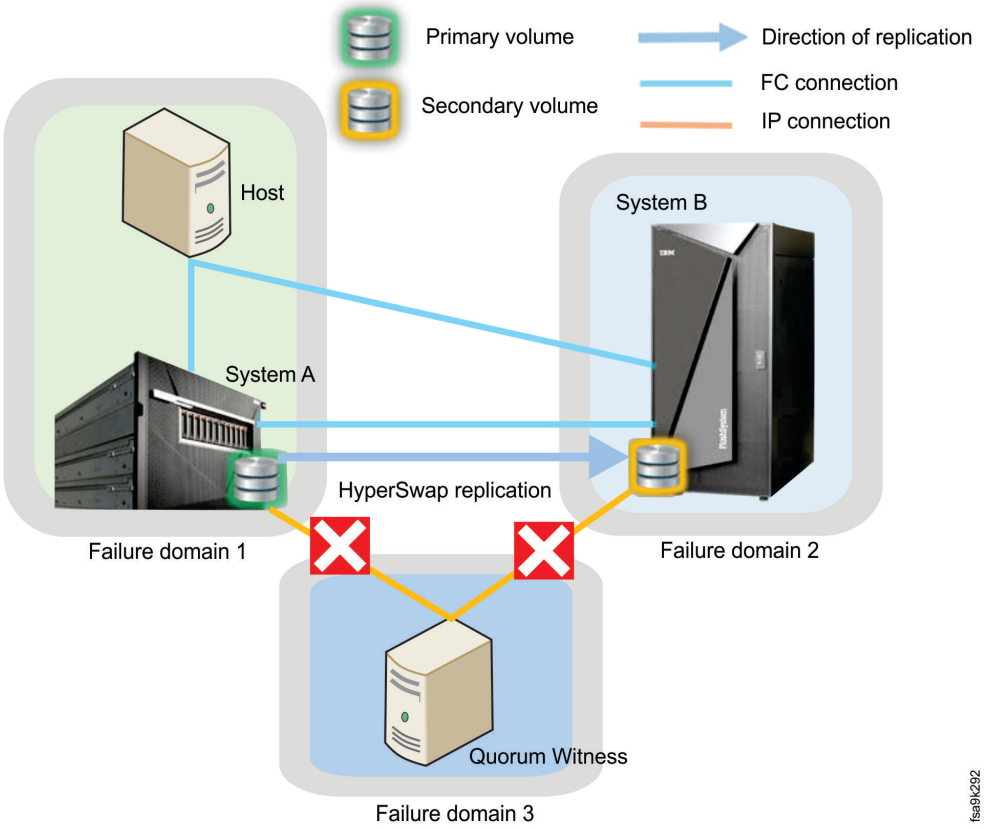


Figure 50. System A - Quorum Witness and System B - Quorum Witness connectivity failure

As soon as connectivity with the Quorum Witness is restored, both peers will automatically resume sending heartbeat messages, and the automatic failover will be automatically resumed.

System A I/O serving failure

If System A is functional, but is unable to serve I/O for any reason, it notifies System B of the failure, locks its HyperSwap volume, and sets the state of its port groups to *Unavailable*, thus blocking even I/O in flight. System B then automatically takes over the Primary volume and starts serving I/O.

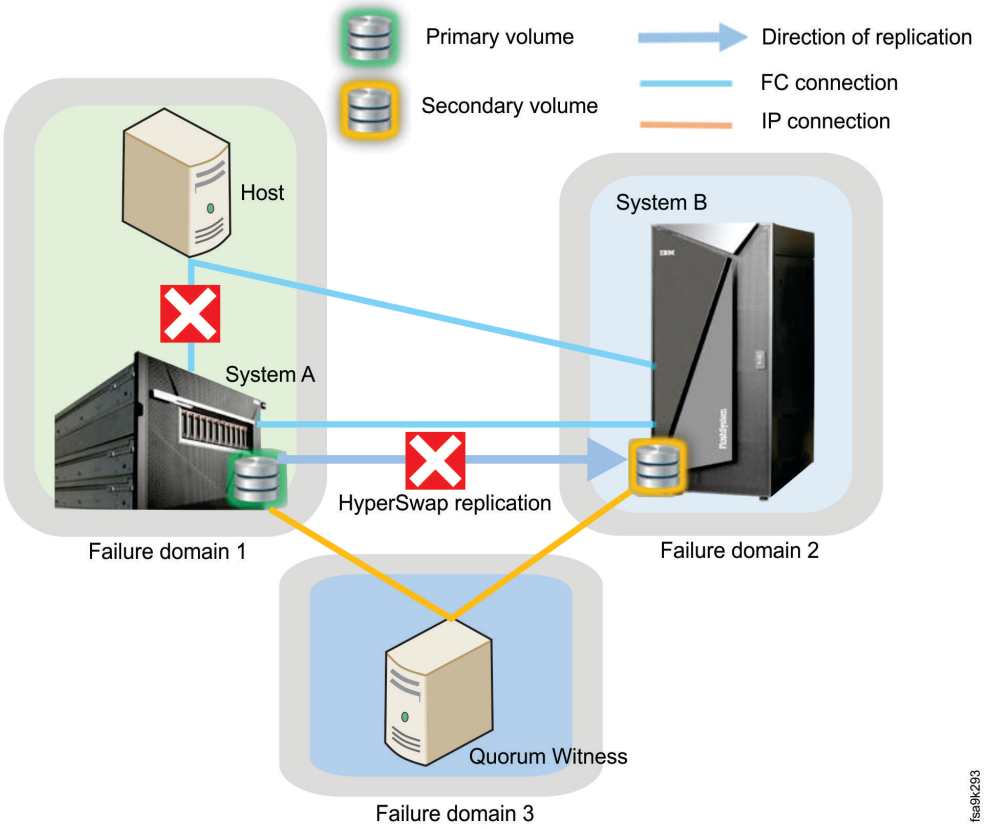


Figure 51. System A I/O serving failure

This scenario requires manual recovery. System A role must be changed to Secondary, and the HyperSwap relationship needs to be activated. At this point data synchronization will begin. After the data is synchronized, the roles can be switched in order to restore the original configuration.

Total connectivity failure

This scenario is very unlikely and practically impossible, if the storage arrays and Quorum Witness are located in separate sites.

If all connectivity between System A, System B, and Quorum Witness is disrupted, none of the peers is aware of its counterpart's state. Therefore, both assume that the counterpart has taken over the volume, and stop serving I/O. The split-brain scenario is thus avoided, and data access will be completely blocked to avoid any data inconsistency.

This scenario requires manual recovery. Peer roles need to be reassigned as necessary, and automatic failover needs to be established.

Chapter 14. Migrating data

The data migration feature enables the transfer of large amounts of data with continuous data access during the process.

The use of any new storage system frequently requires the transfer of large amounts of data from another storage system. The data migration feature enables continuous data access during the actual data transfer, thus reducing the required downtime time to the minimum, sometimes from days or hours to minutes. In addition, data migration is applicable to any source block storage.

The following table summarizes the best practice migration methods.

Table 6. Best practice migration methods

The storage system from which to migrate data	**Best practice migration method**
Non-IBM storage system	Regular data migration
Any IBM storage system, except IBM XIV Gen3, IBM FlashSystem A9000, and FlashSystem A9000R	Regular data migration
IBM XIV Gen3, versions prior to 11.6.2.a	Regular data migration
IBM XIV Gen3, version 11.6.2.a or later	Hyper-Scale Mobility
IBM FlashSystem A9000R, versions prior to 12.2.0	Regular data migration
IBM FlashSystem A9000R version 12.2.0 or later	Hyper-Scale Mobility
IBM FlashSystem A9000	Hyper-Scale Mobility

Note: When the origin system is IBM XIV Gen3 (version 11.6.2.a or later), IBM FlashSystem A9000 (any version), or FlashSystem A9000R (version 12.2.0 or later), the best practice migration method is IBM Hyper-Scale Mobility, since it is non-disruptive (see Chapter 15, “Volume migration with IBM Hyper-Scale Mobility,” on page 101).

Data migration overview

The data migration feature enables the smooth transition of a volume from any block storage to IBM FlashSystem A9000 or A9000R, by:

- Connecting the associated host to IBM FlashSystem A9000 or A9000R and providing the host with direct access to the most up-to-date data even before data has been copied from the origin storage system.
- Transparently copying the contents of the origin storage system to IBM FlashSystem A9000 or A9000R as a background process.

During data migration, the host is connected directly to IBM FlashSystem A9000 or A9000R and is disconnected from the origin storage system to avoid a data integrity issue. IBM FlashSystem A9000 or A9000R is connected to the origin storage system, as shown in the Data migration topology chart below.

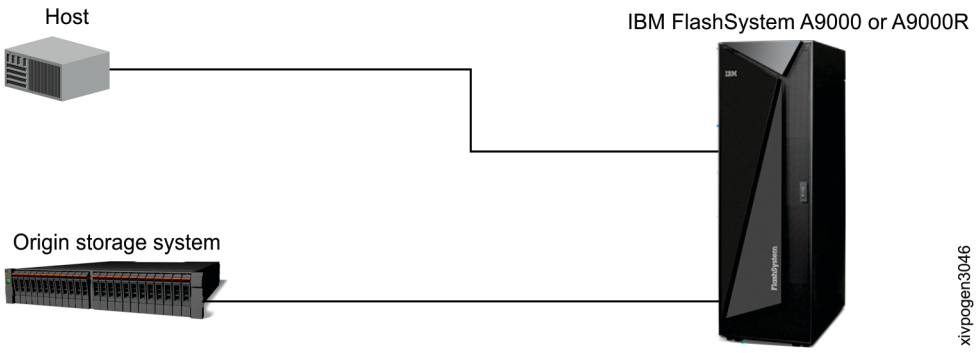


Figure 52. Data migration topology

The communication between the host and IBM FlashSystem A9000 or A9000R and the communication between IBM FlashSystem A9000 or A9000R and the origin storage system is over a Fibre Channel or iSCSI connection.

Important: SCSI Hardware accelerated locking (ATS command) is not supported for data migration. When host operating systems that use ATS are involved, customers should either disable ATS usage on these hosts for the duration of the migration, or consider other migration methods.

Customers who can use VMware storage vMotion for the required volume migration, should prefer this method for two reasons:

- Storage vMotion does not disrupt the involved VMs.
- While valid, disabling ATS (for VAAI-capable hosts) has operational ramifications as described in the VMware Knowledge Base.

I/O handling in data migration

Serving read requests

During this stage, IBM FlashSystem A9000 or A9000R will serve all the host's data read requests. This will be performed in a transparent manner without requiring any action by the host, as follows:

- If the requested data has already been copied to IBM FlashSystem A9000 or A9000R, it is served from IBM FlashSystem A9000 or A9000R.
- If the requested data has not yet been copied, IBM FlashSystem A9000 or A9000R will retrieve it from the origin storage system and then serve it to the host.

Serving write requests

During this stage IBM FlashSystem A9000 or A9000R will serve all host's data write requests. This will be performed in a transparent manner without requiring any action by the host.

Data migration provides the following two alternative configurations for handling write requests from a host:

Source updating:
A host's write requests are written by IBM FlashSystem A9000 or A9000R to IBM FlashSystem A9000 or A9000R, as well as to the origin storage

system. In this case, the origin storage system remains completely updated during the background copying process. Throughout the process, the volume of the origin storage system maintains an up-to-date copy of the data.

Write commands are performed synchronously, so IBM FlashSystem A9000 or A9000R only acknowledges the write operation after writing to itself, writing to the origin storage system, and receiving an acknowledgment from the origin storage system. Furthermore, if, due to a communication error or any other error, the writing to the origin storage system fails, IBM FlashSystem A9000 or A9000R will report to the host that the write operation has failed.

Source updating is the recommended method for handling write requests from the host, because it provides the ability to fall back if the data migration process cannot be successfully completed.

No source updating:

A host's write requests are only written by IBM FlashSystem A9000 or A9000R to IBM FlashSystem A9000 or A9000R and are not written to the origin storage system. In this case, IBM FlashSystem A9000 or A9000R is not updated during the background copying process. Therefore, the two storage systems will never be identical once a write was done by the host to IBM FlashSystem A9000 or A9000R. The volume of the origin storage system will remain intact and will not be changed throughout the data migration process.

Data migration stages

Figure 53 on page 98 depicts the process of migrating a volume from a previous storage system to the new storage system. It also shows how the new storage system synchronizes its data with the previous storage system, and how it handles the data requests of a host throughout all these stages of synchronization.

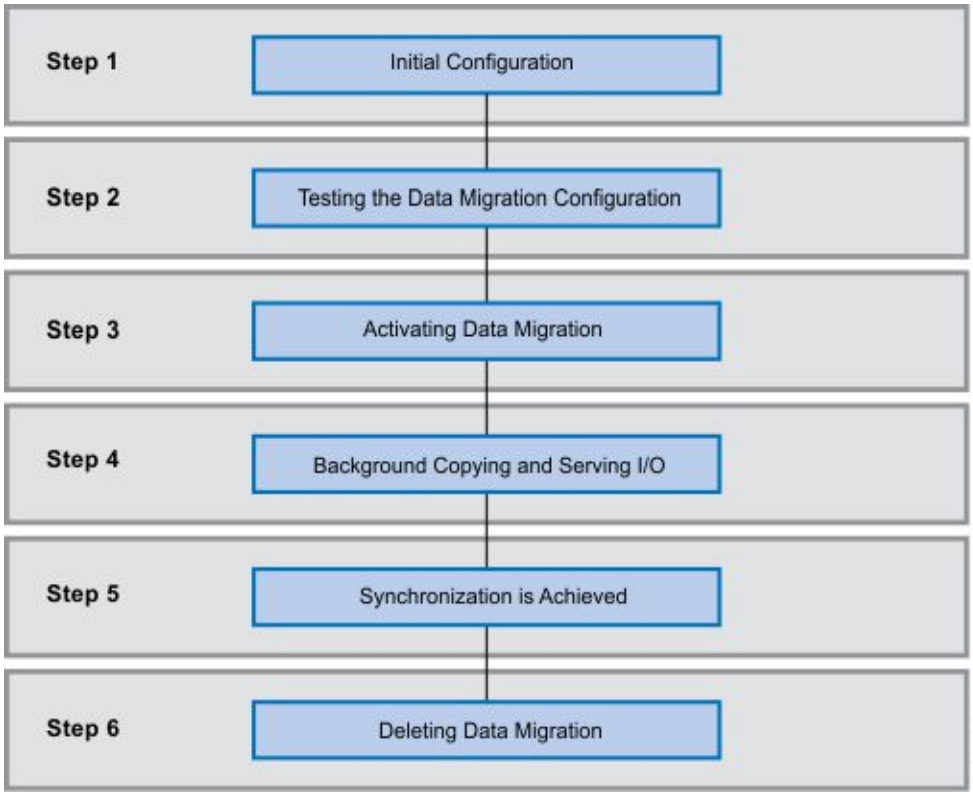


Figure 53. Data migration steps

Initial configuration

The following prerequisite actions must be performed prior to initiating data migration:

- Format the volume on the target IBM FlashSystem A9000 or A9000R.
- Disconnect the actual host from the origin storage system (see "Testing the data migration configuration" for details).
- Connect IBM FlashSystem A9000 or A9000R to the origin storage system whose data it will be serving. Because the origin storage system perceives IBM FlashSystem A9000 or A9000R as a host, IBM FlashSystem A9000 or A9000R must connect to the origin storage system as a SCSI initiator.
- Make sure that the volumes on the origin storage system and on IBM FlashSystem A9000 or A9000R have an equal number of 512 byte blocks. This is verified upon activation of the data migration process.

You can then initiate data migration and configure all hosts to work directly and solely with IBM FlashSystem A9000 or A9000R.

Testing the data migration configuration

Before connecting the host to IBM FlashSystem A9000 or A9000R, test the data migration definitions to verify that IBM FlashSystem A9000 or A9000R can access the origin storage system.

As a mandatory prerequisite for this stage, the actual host must be disconnected from IBM FlashSystem A9000 or A9000R (see "Initial configuration"). Alternatively, one test can be performed while the host is still connected, and another test can be

performed after disconnecting the host. This allows the customer to ensure that the original volume size has not changed.

Activating data migration

After you have tested the connection between IBM FlashSystem A9000 or A9000R and the origin storage system, activate data migration and connect the host to IBM FlashSystem A9000 or A9000R. From this point forward, the host reads and writes data to IBM FlashSystem A9000 or A9000R, and IBM FlashSystem A9000 or A9000R will read and optionally write to the origin storage system.

Once activated, deactivating data migration will cause host I/O to fail. Use deactivation only for recovery after a failure, if the data migration configuration cannot be recovered.

Background copying and serving I/O operations

Once data migration is initiated, it will start a background process of sequentially copying all the data from the origin storage system to IBM FlashSystem A9000 or A9000R.

Synchronization is achieved

After all of a volume's data has been copied, the data migration achieves synchronization. After synchronization is achieved, all read requests are served from IBM FlashSystem A9000 or A9000R.

If source updating is enabled, IBM FlashSystem A9000 or A9000R will continue to write new data to both itself and the origin storage system until data migration settings are deleted (for details on source updating, see “Serving write requests” on page 96.)

Deleting data migration

When FlashSystem A9000 or A9000R is synchronized with the origin storage system, the data migration process can be safely deactivated and deleted together with the source volume.

If the data migration configuration is incorrect (for example, a wrong source volume is selected or the defined data block size is unsuitable) or if data migration is activated merely for a proof of concept, the process can be forcefully terminated, deactivated, and deleted. In this case, even if the target volume may seem redundant, make sure that any written data is preserved before deleting it. This is especially important if source updating was not activated during data migration, because the target volume may contain data that is missing from the source volume.

If source updating was activated during data migration, all written data will remain on the source volume.

Handling failures

Upon a communication error or the failure of the origin storage system, I/Os received on IBM FlashSystem A9000 or A9000R are likely to fail as source updating writes cannot be sent to the origin storage system, and data not yet copied to IBM FlashSystem A9000 or A9000R cannot be read.

If IBM FlashSystem A9000 or A9000R encounters a Medium Error on the origin storage system (meaning that IBM FlashSystem A9000 or A9000R cannot read a data block on the origin storage system), then the state of the same data block in IBM FlashSystem A9000 or A9000R will indicate a Medium Error, even though the backend persistent media in IBM FlashSystem A9000 or A9000R have not failed.

Chapter 15. Volume migration with IBM Hyper-Scale Mobility

IBM Hyper-Scale Mobility enables a non-disruptive migration of volumes from IBM FlashSystem A9000 or A9000R to IBM FlashSystem A9000 or A9000R storage system, and from XIV Gen3 to IBM FlashSystem A9000 or A9000R.

IBM Hyper-Scale Mobility dramatically minimizes time, complexity, risk, cost and variety of tools required to non-disruptively vacate complete storage arrays, that serve different host operating systems and applications. Typically, it helps achieve data migration in the following scenarios:

- Migrating data out of an over-provisioned system.
- Migrating all the data from a system that will be decommissioned or re-purposed.
- Migrating data to another storage system to achieve adequate (lower or higher) performance, or to load-balance systems to ensure uniform performance.
- Migrating data to another storage system to load-balance capacity utilization.

Note: Starting from version 12.2.1, customers can non-disruptively move applications out of multiple XIV Gen3 storage systems and them consolidate them onto a single IBM FlashSystem A9000 or IBM FlashSystem A9000R.

Figure 54. Cross-generation volume migration

For support information on IBM Hyper-Scale Mobility, see the feature availability matrix.

Important: When using IBM Hyper-Scale Mobility on Storage Area Network (SAN) Boot Volumes, the Proxy targets need to be replaced with the Owner targets on the adapter BIOS. In some cases, this can be done online from the operating system, by using vendor's online tools for x86 servers (for example, Qlogic QConvergeConsole), or by using specific commands to edit the boot paths order for Unix servers. In other cases, manual changes of the adapter BIOS are required.

When migrating volumes from a storage system that does not support IBM Hyper-Scale Mobility, a best practice is to use the regular data migration. It minimizes the duration of the disruption by concurrently allowing both data access and data migration.

Volume migration process

IBM Hyper-Scale Mobility moves a volume from one system to another, while the host keeps using the volume.

To accomplish this, I/O paths are manipulated by the storage, without involving host configuration, and the volume identity is cloned on the target system. In addition, direct paths from the host to the target system need to be established, and paths to the original host can finally be removed. Host I/Os are not interrupted throughout the migration process.

The key stages of the IBM Hyper-Scale Mobility and the respective states of volumes are shown in Figure 55 on page 103 and explained in detail in Table 7 on page 103.

For an in-depth practical guide to using IBM Hyper-Scale Mobility, refer to the following Redpaper™ publication: IBM Hyper-Scale Mobility Overview and Usage.

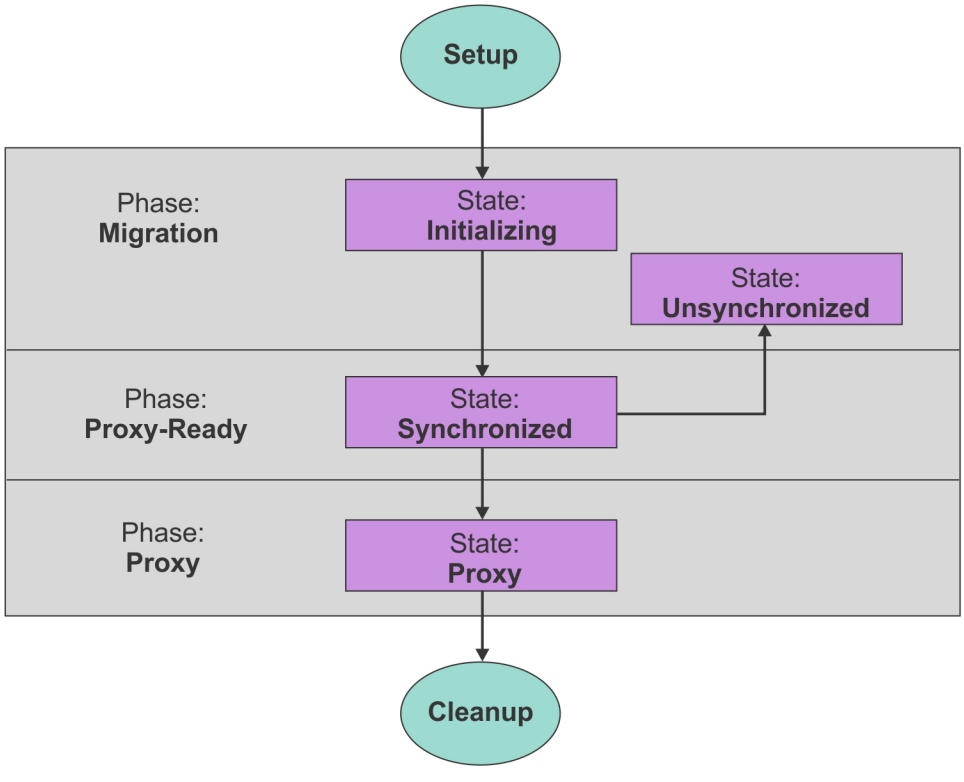


Figure 55. Volume migration flow of the IBM Hyper-Scale Mobility

Table 7. Volume migration stages

Stage	Description	Source and destination volume states
Setup	A volume is automatically created at the destination storage system with the same name as the source volume. The relation between the source and destination volumes is established.	The two volumes are not yet synchronized.
Migration	New data is written to the source and replicated to the destination.	**Initializing** - The content of the source volume is copied to the destination volume. The two volumes are not yet synchronized. This state is similar to the Initialization state of synchronous mirroring. As long as the source instance cannot confirm that all of the writes were acknowledged by the destination volume, the state remains Initializing.

Table 7. Volume migration stages (continued)

Stage	Description	Source and destination volume states
Proxy-Ready	The replication of the source volume data is complete when the destination is synchronized. The source serves host writes as a proxy between the host and the destination. The system administrator issues a command that moves the volume migration relation to the proxy. Next, the system administrator maps the host to the destination. In this state, a single copy of the data exists on the destination and any I/O directed to the source is redirected to the destination.	**Synchronized** - The source was fully copied to the destination. This state is similar to the Synchronized state of synchronous mirroring.
Proxy	New data in written to the source and is migrated to the destination. The proxy serves host requests as if it were the target, but it actually impersonates the target. **Note:** It is a best practice for the IBM Hyper-Scale Mobility to remain in Proxy state for a minimal period.	**Proxy** - The source acts as a proxy to the destination.
Cleanup	After validating that the host has connectivity to the destination volume through the new paths, the storage administrator unmaps the source volume on the source storage system from the host. Then the storage administrator ends the proxy and deletes the relationship.	

Chapter 16. Data-at-rest encryption

The storage system secures all written data with industry-standard AES-256 encryption for data-at-rest.

Data-at-rest encryption protects the grid controller SSD disks and flash enclosure MicroLatency modules against exposure of sensitive data on discarded or stolen media by ensuring that the data cannot be read as long as the key used to encrypt the data is secured.. Encryption is carried out at the hardware level to avoid any performance impact.

Each MicroLatency module has FPGA control for data-at-rest encryption. Data-at-rest is protected by an Advanced Encryption Standard (XTS-AES) algorithm using the 256-bit symmetric option in xor-encrypt-xor (XEX)-based tweaked-codebook mode with ciphertext stealing (XTS) mode, as defined in the IEEE1619-2007 standard.

In addition, an HMAC-SHA256 algorithm is used to create a hash message authentication code (HMAC) for corruption detection. A system-generated cyclic redundancy check (CRC) is applied for additional protection.

Encryption can be enabled during the installation of the system or at any time later. While encryption is not enabled, the system might not meet customers or legal compliance standards and the data might not be protected against security issues. Encryption can be disabled only when no volumes are defined.

Encryption key management schemes

Encryption key management can be carried out through an external or an internal scheme.

External encryption key management scheme

With an external encryption key management scheme, encryption keys are stored separately from the data, thereby presenting a secured and well-defined interface for key services.

The separation of key storage from data storage and key management is accomplished with external Key Management Interoperability Protocol (KMIP) compliant servers, such as IBM Security Key Lifecycle Manager (SKLM) or Gemalto SafeNet KeySecure server.

The supported versions of IBM Security Key Lifecycle Manager (SKLM) are 2.6 and 2.7.

The following table shows Gemalto SafeNet KeySecure server versions supported by various versions of IBM FlashSystem A9000 and A9000R.

Table 8. Gemalto SafeNet KeySecure server support

IBM FlashSystem A9000 and A9000R version	Supported versions of Gemalto SafeNet KeySecure server
12.0 - 12.0.2	8.3.2 or earlier

Table 8. Gemalto SafeNet KeySecure server support (continued)

IBM FlashSystem A9000 and A9000R version	Supported versions of Gemalto SafeNet KeySecure server
12.0.3	8.4 or earlier
12.2.1.b and later	8.6 - 8.9

Important: The versions of IBM FlashSystem A9000 and A9000R from version 12.1.0 and before version 12.2.1.b do not support Gemalto SafeNet KeySecure server.

Note: Future versions of Gemalto SafeNet KeySecure server may also be supported.

SKLM is also used to locally create, distribute, backup, and manage the life cycle of keys and certificates.

Internal encryption key management scheme

The internal encryption key management scheme does not warrant the purchase, deployment, or management of a dedicated, independent key management system, because the encryption key is generated and stored within the storage system. In addition, with an encryption internal encryption key management scheme, keys are not affected by software upgrades, and remain available upon the failure of up to two grid controllers.

Chapter 17. User roles and permissions

User roles allow specifying which roles are applied and the various applicable limits.

Note: None of these system-defined users have access to data.

Table 9. User roles and permissions

User role	Permissions and limits	Typical usage
Read-only	Read-only users can only list and view system information.	The system operator, typically, but not exclusively, is responsible for monitoring system status and reporting and logging all messages.
Application administrator	Only application administrators carry out the following tasks: • Creating snapshots of assigned volumes • Mapping their own snapshot to an assigned host • Deleting their own snapshot	Application administrators typically manage applications that run on a particular server. Application managers can be defined as limited to specific volumes on the server. Typical application administrator functions: • Managing backup environments: – Creating a snapshot for backups – Mapping a snapshot to back up server – Deleting a snapshot after backup is complete – Updating a snapshot for new content within a volume • Managing software testing environment: – Creating an application instance – Testing the new application instance
Security administrator	Carries out encryption-related tasks and does not have to be a storage administrator.	All encryption-related operations.

Table 9. User roles and permissions (continued)

User role	Permissions and limits	Typical usage
Storage administrator	Has permission to all functions, except: • Maintenance of physical components or changing the status of physical components • Only the predefined administrator, named *admin*, can change the passwords of other users A predefined storage administrator account is provided with the system and cannot be deleted.	Storage administrators are responsible for all administration functions.
Domain administrator	Can administer a specific domain. For more information, see Chapter 19, "Multi-tenancy," on page 115.	When using the multi-tenancy feature.
Technician	The technician is limited to the following tasks: • Physical system maintenance • Phasing components in or out of service A predefined technician account is provided with the system and cannot be deleted.	Technicians maintain the physical components of the system. Only one predefined technician is specified per system. Technicians are IBM technical support team members.

Note:

1. All users can view the status of physical components; however, only technicians can modify the status of components.
2. User names are case-sensitive.
3. Passwords are case-sensitive.

User groups

A user group is a group of application administrators who share the same set of snapshot creation permissions. This enables a simple update of the permissions of all the users in the user group by a single command.

The permissions are enforced by associating the user groups with hosts or clusters. User groups have the following characteristics:

- Only users who are defined as application administrators can be assigned to a group.
- A user can belong to only a single user group.
- A user group can contain up to eight users.
- If a user group is defined with `access_all=yes`, application administrators who are members of that group can manage all volumes on the system.

Storage administrators create the user groups and control the various permissions of the application administrators.

Hosts and clusters can be associated with only a single user group. When a user belongs to a user group that is associated with a host, it is possible to manage snapshots of the volumes mapped to that host. User and host associations have the following properties:

- User groups can be associated with both hosts and clusters. This enables limiting application administrator access to specific volumes.
- A host that is part of a cluster cannot also be associated with a user group.
- When a host is added to a cluster, the associations of that host are broken. Limitations on the management of volumes mapped to the host is controlled by the association of the cluster.
- When a host is removed from a cluster, the associations of that host become the associations of the cluster. This enables continued mapping of operations so that all scripts will continue to work.

Predefined users

Two administrative users are preconfigured on the storage system at the factory. These users cannot be deleted.

Storage administrator
This user provides the highest level of customer access to the system.

Predefined user name: `admin`

Default password: `adminadmin`

Technician
This user is only for IBM service personnel.

Predefined user name: `technician`

Default password: Password is predefined and is used only by the IBM technicians.

Note: Predefined users are always authenticated by the storage system, even if LDAP authentication has been activated for them.

User information

Configuring users requires defining the following options:

Role Specifies the role category that each user has when operating the system. The role category is mandatory (see Chapter 17, "User roles and permissions," on page 107).

Name Specifies the name of each user allowed to access the system.

Password
All user-definable passwords are case sensitive.
Passwords are mandatory, can be 6 to 12 characters long, use uppercase or lowercase letters as well as the following characters: `~!@#$%^&*()_+-={}|:;<>?,./\[]` .

Email Email is used to notify specific users about events through e-mail messages. Specifying the user's email address is optional and must adhere to the standard email format.

Phone and area code

Phone numbers are used to send SMS messages to notify specific users about events. Phone number and area code stings can contain up to 63 digits, including hyphens (-) and periods (.)

Chapter 18. User authentication and access control

The storage system features role-based authentication either natively or by using LDAP-based authentication.

The system provides:

Role-based access control
Built-in roles for access flexibility and a high level of security according to predefined roles and associated tasks.

Two methods of access authentication
The following methods of user authentication are supported:

Native authentication
This is the default mode for authentication of users and groups that are defined on the storage system. In this mode, users and groups are authenticated against a database on the system.

LDAP When enabled, the system authenticates the users against an LDAP repository.

Note: The administrator and technician roles are always authenticated by the storage system, regardless of the authentication mode.

Native authentication

Native authentication if the default mode for authenticating users and user groups.

In this mode, users and groups are authenticated against a database on the system, based on the submitted username and password, which are compared to user credentials defined and stored on the storage system.

The authenticated user must be associated with a user role that specifies the system access rights.

LDAP authentication

Lightweight Directory Access Protocol (LDAP) support enables the authentication of storage system users through an LDAP server repository, or directory.

When LDAP authentication is enabled, the username and password of a user accessing the storage system (through the CLI or GUI) are used by the system to log in to a specified LDAP directory. Upon a successful login, the storage system retrieves the user's group membership data stored in the LDAP directory, and then uses that information to associate the user with a storage system administrative role.

The group membership data is stored in a customer defined, preconfigured attribute on the LDAP directory. This attribute contains string values which are associated with administrative roles. These values might be LDAP Group Names, but this is not required by the storage system. The values the attribute contains, and their association with storage system administrative roles, are also defined by the customer.

Supported domains

The following LDAP authentication directories are supported:

- Microsoft Active Directory
- SUN directory
- Open LDAP

LDAP multiple-domain implementation

In order to support multiple LDAP servers that span over different domains, and in order to use the **memberOf** property, the storage system allows for more than one role for the Storage Administrator and the Read⌂Only roles.

The predefined administrative IDs "**admin**" and "**technician**" are always authenticated by the storage system, whether or not LDAP authentication is enabled.

Responsibilities division between the LDAP directory and the storage system

LDAP and the storage system divide responsibilities and maintained objects.

Following are responsibilities and data maintained by the storage system and the LDAP directory:

LDAP directory

- Responsibilities: user authentication for system users, and assignment of a system-related user group.
- Maintains: users, username, password, designated system-related LDAP groups.

Storage system

- Responsibilities: determining the appropriate user role by mapping the LDAP user group to a storage system role, and enforcement of any user access to the storage system.
- Maintains: mapping of the LDAP user group to a storage system role.

LDAP authentication logic

The LDAP authentication procedure consists of several key steps.

1. The LDAP server and system parameters must be defined.
2. A storage system user must be defined on that LDAP server. The storage system uses this user when searching for authenticated users. This user is later on referred to as system's configured service account.
3. The LDAP user requires an attribute in which the values of the storage system user roles are stored.
4. Mapping between LDAP user attributes and storage system user roles must be defined.
5. LDAP authentication must be enabled on the storage system.

Once LDAP is configured and enabled, the predefined user is granted with login credentials authenticated by the LDAP server, rather than the storage system itself.

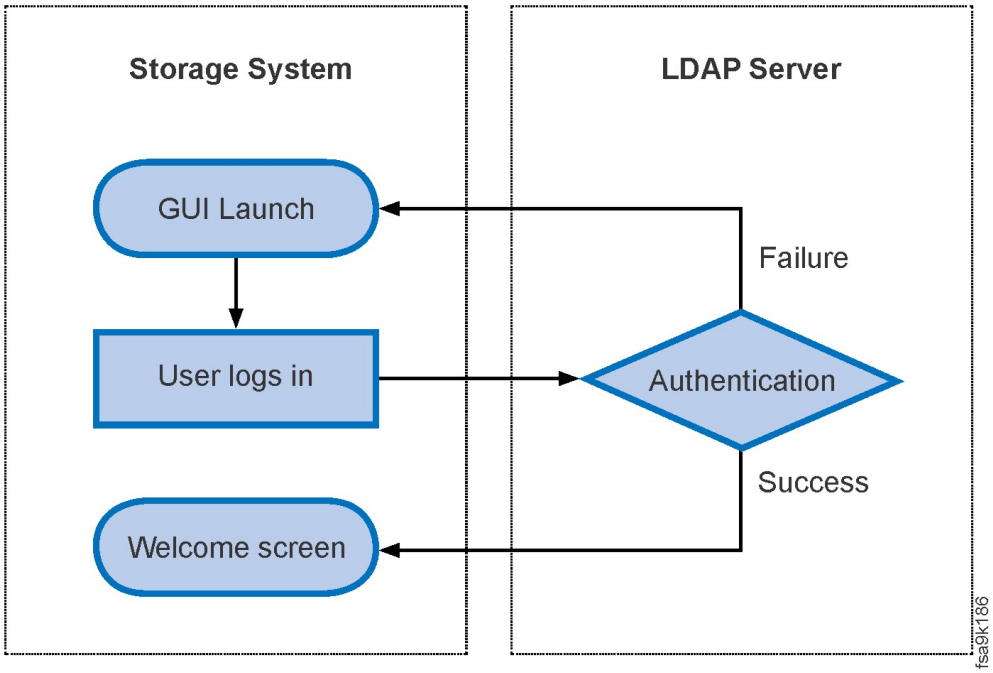


Figure 56. Login to a specified LDAP directory

The storage administrator can test the LDAP configuration before its activation.

User validation

During the login, the system validates the LDAP user as follows:

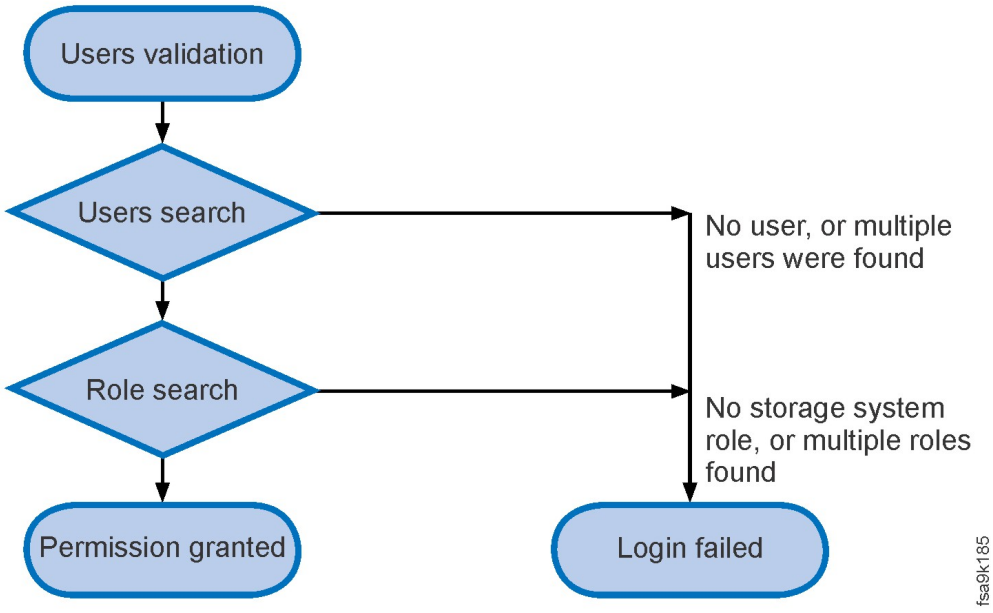


Figure 57. The way the system validates users through issuing LDAP searches

Issuing a user search

The system issues an LDAP search for the user's entered username. The request is submitted on behalf of the system's configured service account

and the search is conducted for the LDAP server, base DN and reference attribute as specified in the storage system LDAP configuration.

The base DN specified in the storage system LDAP configuration serves as a reference starting point for the search – instructing LDAP to locate the value submitted (the username) in the attribute specified.

If a single user is found - issuing a storage system role search
The system issues a second search request, this time submitted on behalf of the user (with the user's credentials), and will search for storage system roles associated with the user, based on the storage system LDAP configuration settings.

- If a single storage system role is found - permission is granted
The system inspects the rights associated with that role and grant login to the user. The user's permissions are in correspondence with the role associated by the storage system, base on the storage system LDAP configuration.

- If no storage system role is found for the user, or more than one role was found
If the response by LDAP indicates that the user is either not associated with a storage system role (no user role name is found in the referenced LDAP attribute for the user), or is actually associated with more than a single role (multiple roles names are found) – login will fail and a corresponding message will be returned to the user.

If no such user was found, or more than one user were found
If LDAP returns no records (indicating no user with the username was found) or more than a single record (indicating that the username submitted is not unique), the login request fails and a corresponding message is returned to the user.

Chapter 19. Multi-tenancy

The storage system allows allocating storage resources to several independent administrators, assuring that one administrator cannot access resources associated with another administrator.

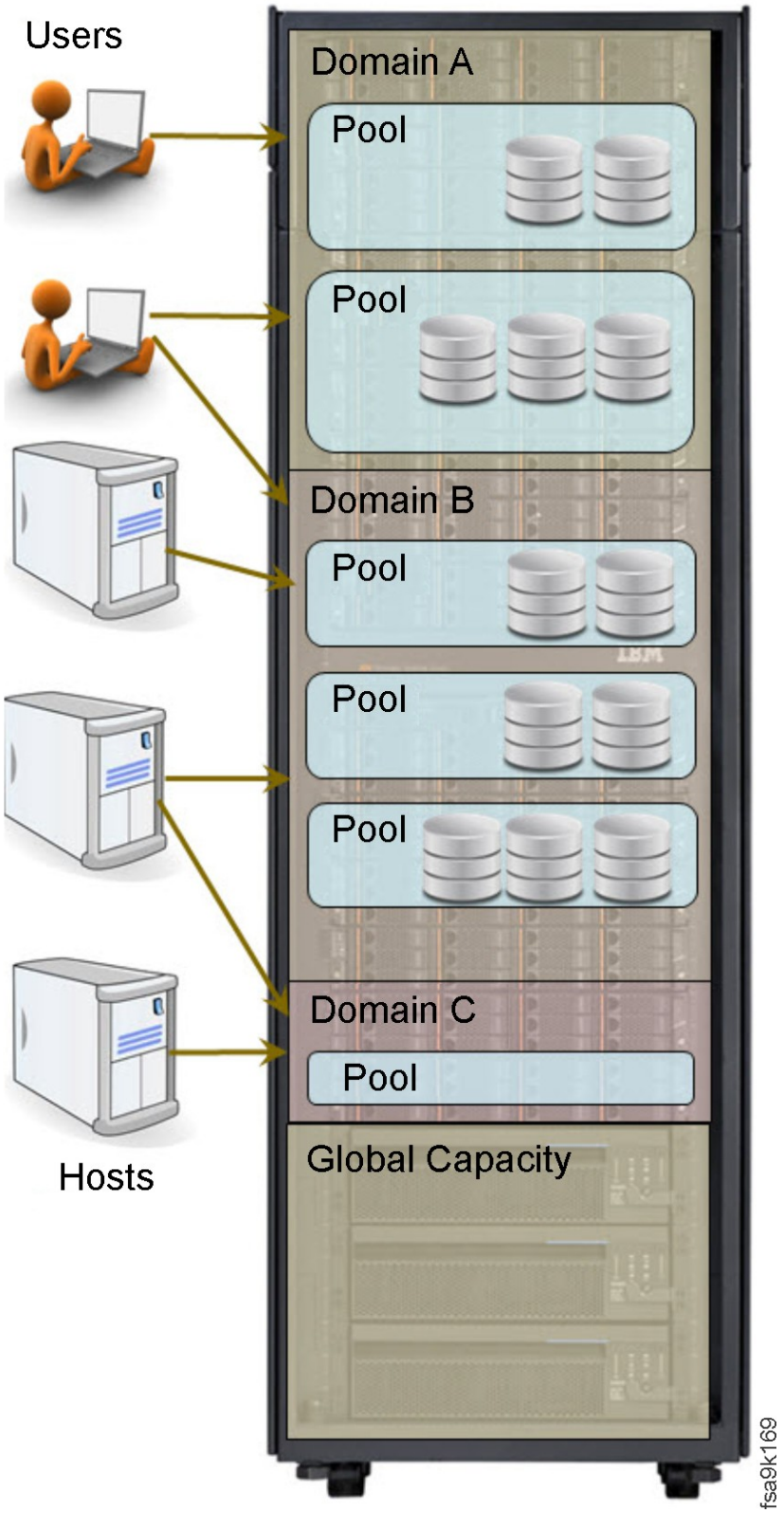
Multi-tenancy extends the storage system approach to role-based access control. In addition to associating the user with predefined sets of operations and scope (the applications on which an operation is allowed), the storage system enables the user to freely determine what operations are allowed, and where they are allowed.

When multi-tenancy is enabled, storage resources are allocated to several independent administrators with the assurance that one administrator cannot view or access resources that are associated with another administrator.

This resource allocation is a partitioning of the system's resources to separate administrative *domains*.

A *domain* is a subset, or partition, of the system's resources. It is a named object to which users, storage pools, hosts, clusters, targets, and other entities can be associated. The domain restricts the resources that a user can manage to those associated with the domain.

A domain maintains the user relationships that exist at the storage system level, as shown in the following figure.



A *domain administrator* is the user who is associated with a domain and is able to manage it. The domain administrator is restricted to performing operations only on objects that are within the specific domain to which the administrator is associated.

The following access rights and restrictions apply to domain administrators:

- A user is created and assigned a role (for example: storage administrator, application administrator, read-only).
- When assigned to a domain, the user retains his given role, limited to the scope of the domain.
- Access to objects in a domain is restricted up to the point where the defined user role intersects the specified domain access.
- By default, domain administrators cannot access objects that are not associated with their domains.
- The domain is an isolated set of storage resources.
- The domain administrator has access only to the specified domains.
- The global administrator can manage domains and assign administrators to domains.
- Private objects are assigned to domains.
- The domain maintains its connectivity to global objects, such as: users, hosts, clusters, and targets. Hosts and clusters can serve several domains. However, hosts created by a domain administrator are assigned only to that domain.

Multi-tenancy offers the following benefits:

- **Partitioning** – The system resources are partitioned to separate domains. The domains are assigned to different tenants and each tenant administrator gets permissions for a specific, or several domains, to perform operations only within the boundaries of the associated domain(s).
- **Self-sufficiency** – The domain administrator has a full set of permissions needed for managing all of the domain resources.
- **Isolation** – There is no visibility between tenants. The domain administrator is not informed of resources outside the domain. These resources are not displayed on lists, nor are their relevant events or alerts displayed.
- **User-domain association** – A user can have a domain administrator role on more than one domain.
- **Users other than the domain administrator** – Storage, security, and application administrators, as well as read-only users, retain their right to perform the same operations that they have in a non-domain-based environment. These users can access the same objects under the same restrictions.
  - **Global administrator** – Not associated with any specific domain, and determines the operations that can be performed by the domain administrator in a domain. This is the only user that can create, edit, and delete domains, and associate resources to a domain.

    An *open* or *closed* policy can be defined so that a global administrator may, or may not, be able to see inside a domain.

    Intervention of a global domain administrator, that has permissions for the global resources of the system, is only needed for:

    - Initial creation of the domain and assigning a domain administrator.
    - Resolving hardware issues.
  - **Global user** – A user that is not associated with any domain has access rights to all of the entities that are not uniquely associated with a domain.

Working with multi-tenancy

This section provides a general description about working with multi-tenancy and its attributes.

Domain administrator assignment and operations

The domain administrator assignment and permitted operations are as follows:

- Prior to its association with a domain, the future domain administrator (currently a system administrator) has access to all non-domain entities, and no access to domain-specific entities.
- When the storage administrator becomes a domain administrator, all access rights to non-domain entities are lost.
- The domain administrator can map volumes to hosts as long as both the volume and the host belong to the domain.
- The domain administrator can copy and move volumes across pools as long as the pools belong to domains administered by the domain administrator.
- Domain administrators can manage snapshots for all volumes in their domains.
- Domain administrators can manage consistency and snapshot groups for all pools in their domains. Moving consistency groups across pools is allowed as long as both source and destination pools are in the administrator's domains.
- Domain administrators can create and manage pools under the storage constraint associated with their domain.
- Although not configurable by the domain administrator, hardware list, and events - are available for view-only to the domain administrator within the scope of the domain.
- Commands that operate on objects not associated with a domain are not accessible by the domain administrator.

Domain attributes

The domain has the following attributes:

- **Capacity** – the domain is allocated with a capacity that is further allocated among its pools. The domain provides an additional container in the hierarchy of what was once *system-pool-volume*, and is now *system-domain-pool-volume*:
  - The unallocated capacity of the domain is reserved to the domain storage pools.
  - The sum of the physical capacity of the system's domains cannot exceed the storage system total physical capacity.
  - The sum of allocation limits of the system's domains cannot exceed the storage system maximum allocation limit.
- **Maximum number of volumes per domain** – the maximum number of volumes per system is divided among the domains in a way that one domain cannot consume all of the system resources at the expense of the other domains.
- **Maximum number of pools per domain** – the maximum number of pools per system is divided among the domains in a way that one domain cannot consume all of the system resources at the expense of the other domains.
- **Maximum number of mirrors per domain** – the maximum number of mirrors per system is divided among the domains.
- **Maximum number of consistency groups per domain** – the maximum number of consistency groups per system is divided among the domains.
- **Performance class** – the maximum aggregated bandwidth and IOPS is calculated for all volumes of the domain, rather than on a system level.
- The domain has a string that identifies it for LDAP authentication.

Mirroring a multi-tenancy environment

- The target, target connectivity and interval schedule are defined, edited and deleted by the storage administrator.
- The domain administrator can create, activate and change properties to a mirroring relation based on the previously defined target and target connectivity that are associated with the domain.
- The remote target does not have to belong to a domain.
- Whenever the remote target belongs to a domain, it checks that the remote target, pool and volume (if specified upon the mirror creation) all belong to the same domain.

Chapter 20. Management and monitoring

The storage system can be monitored and fully controlled by using different management and automation tools.

The primary management tools for storage administrators are:

- **IBM Hyper-Scale Manager** – Management server software connects to and controls one or more storage systems. Remote users can log into the server and use its advanced web-based user interface (UI) for managing and monitoring multiple storage systems in real time.

  The web UI can be accessed in a web browser pointing to the web address of the Hyper-Scale Manager server, together with the port number that is used to open the connection.

  For information about how to use IBM Hyper-Scale Manager, refer to IBM Hyper-Scale Manager on IBM Knowledge Center (ibm.com/support/knowledgecenter/STJKN5_12.2.1/fs9kr_kc_hsm_managing.html).
- **IBM XCLI Utility** – Provides a terminal-based command-line interface for issuing storage system management, monitoring, and maintenance commands from a client computer upon which the utility is installed.

  The command-line interface is a comprehensive, text-based tool that is used to configure and monitor the system. Commands can be issued to configure, manage, or maintain the system, including commands to connect to hosts and applications.

  For information on how to use IBM XCLI Utility, refer to the CLI Reference Guide (ibm.com/support/knowledgecenter/STJKN5_12.2.1/fs9kr_kc_cli_reference.html).

Programmers can utilize the system's advanced application programming interfaces (APIs) for controlling and automating the system:

- Representational state transfer (REST) APIs
- CIM/SMI-S open APIs
- SNMP

Chapter 21. Event reporting and handling

The health, configuration changes, and entire activity of the storage systems are constantly monitored by internal monitoring mechanisms that generate system events.

These events are accumulated by the system and can help the user in the following two ways:

- Users can view past events using various filters. This is useful for troubleshooting and problem isolation.
- Users can configure the system to send one or more notifications, which are triggered upon the occurrence of specific events. These notifications can be filtered according to the events, severity and code. Notifications can be sent through e-mail, SMS messages, or SNMP traps.

Event information

Each event has a predefined structure and is assigned a severity level.

Events are created by various processes, including:

- Object creation or deletion, including volume, snapshot, map, host, and storage pool
- Physical component events
- Network events

Each event contains the following information:

- A system-wide unique numeric identifier
- A code that identifies the type of the event
- Creation timestamp
- Severity
- Related system objects and components, such as volumes, disks, and modules
- Textual description
- Alert flag, where an event is classified as alerting by the event notification rules.
- Cleared flag, where alerting events can be either uncleared or cleared. This is only relevant for alerting events.

Event information can be classified with one of the following severity levels:

Critical
Requires immediate attention

Major Requires attention soon

Minor Requires attention within the normal business working hours

Warning
Non-urgent attention is required to verify that there is no problem

Informational
Normal working procedure event

The storage system provides the following variety of criteria for displaying a list of events:

- Before timestamp
- After timestamp
- Code
- Severity from a certain value and up
- Alerting events, meaning events that are sent repeatedly according to a snooze timer
- Uncleared alerts

The number of displayed filtered events can be restricted.

Event notification rules

The storage system monitors the health, configuration changes, and activity of your storage systems and sends notifications of system events as they occur.

Event notifications are sent according to the following rules:

Which events
The severity, event code, or both, of the events for which notification is sent.

Where The destinations or destination groups to which notification is sent, such as cellular phone numbers (SMS), e-mail addresses, and SNMP addresses.

Notifications are sent according to the following rules:

Destination
The destinations or destination groups to which a notification of an event is sent.

Filter A filter that specifies which events will trigger the sending of an event notification. Notification can be filtered by event code, minimum severity (from a certain severity and up), or both.

Alerting
To ensure that an event was indeed received, an event notification can be sent repeatedly until it is cleared by a CLI command or GUI operation. Such events are called alerting events. Alerting events are events for which a snooze time period is defined in minutes. This means that an alerting event is resent repeatedly each snooze time interval until it is cleared. An alerting event is uncleared when it is first triggered, and can be cleared by the user. The cleared state does not imply that the problem has been solved. It only implies that the event has been noted by the relevant person who takes the responsibility for fixing the problem. There are two schemes for repeating the notifications until the event is clear: snooze and escalation.

Snooze
Events that match this rule send repeated notifications to the same destinations at intervals specified by the snooze timer until they are cleared.

Escalation
You can define an escalation rule and escalation timer, so that if events are not cleared by the time that the timer expires, notifications are sent to the

predetermined destination. This enables the automatic sending of notifications to a wider distribution list if the event has not been cleared.

The following limitations apply to the configuration of alerting rules:

- Rules cannot escalate to nonalerting rules, meaning to rules without escalation, snooze, or both.
- Escalation time should not be defined as shorter than snooze time.
- Escalation rules must not create a loop (cycle escalation) by escalating to itself or to another rule that escalates to it.
- The configuration of alerting rules cannot be changed while there are still uncleared alerting events.

Event notification destinations

Event notifications can be sent to one or more destinations, meaning to a specific SMS cell number, e-mail address, or SNMP address, or to a destination group comprised of multiple destinations.

SMS destination

An SMS destination is defined by specifying a phone number. When defining a destination, the prefix and phone number should be separated because some SMS gateways require special handling of the prefix.

By default, all SMS gateways can be used. A specific SMS destination can be limited to be sent through only a subset of the SMS gateways.

E-mail destination

An e-mail destination is defined by an e-mail address. By default, all SMTP gateways are used. A specific destination can be limited to be sent through only a subset of the SMTP gateways.

SNMP managers

An SNMP manager destination is specified by the IP address of the SNMP manager that is available to receive SNMP messages.

Destination groups

A destination group is simply a list of destinations to which event notifications can be sent. A destination group can be comprised of SMS cell numbers, e-mail addresses, SNMP addresses, or any combination of the three. A destination group is useful when the same list of notifications is used for multiple rules.

Event notification gateways

Some event notifications types require gateway definitions in order to enable the notification delivery.

E-mail (SMTP) gateways

Several e-mail gateways can be defined to enable notification of events by e-mail. By default, the storage system attempts to send each e-mail notification through the first available gateway according to the order that you specify. Subsequent

gateways are only attempted if the first attempted gateway returns an error. A specific e-mail destination can also be defined to use only specific gateways.

All event notifications sent by e-mail specify a sender whose address can be configured. This sender address must be a valid address for the following two reasons:

- Many SMTP gateways require a valid sender address or they will not forward the e-mail.
- The sender address is used as the destination for error messages generated by the SMTP gateways, such as an incorrect e-mail address or full e-mail mailbox.

E-mail-to-SMS gateways

SMS messages can be sent to cell phones through one of a list of e-mail-to-SMS gateways. One or more gateways can be defined for each SMS destination.

Each such e-mail-to-SMS gateway can have its own SMTP server, use the global SMTP server list, or both.

When an event notification is sent, one of the SMS gateways is used according to the defined order. The first gateway is used, and subsequent gateways are only tried if the first attempted gateway returns an error.

Each SMS gateway has its own definitions of how to encode the SMS message in the e-mail message.

Chapter 22. Integration with ISV environments

The storage system can be fully integrated with different independent software vendor (ISV) platforms, APIs, and cloud environments, such as Microsoft Hyper-V, VMware vSphere, OpenStack, and more.

This integration can be implemented natively or by using IBM cloud software solutions, which can facilitate and enhance this integration.

For more information about the available cloud storage solutions, see the '**Platform and application integration**' section on IBM Knowledge Center.

Supporting VMware vStorage extended operations

Extended operations through the VMware vStorage APIs for Array Integration (VAAI) are natively supported, allowing the offload of these operations to the storage system.

The following extended operations are offloaded from the VMware ESXi server to the storage system:

Full copy (clone)
Copies data from one logical unit (LUN based) to another without writing to the ESX server. Rather than issuing read and write requests from the host, the data copying operation is initiated on the storage system. This speeds up the virtual machine (VM) cloning operation and reduces the CPU load on the host.

Block zeroing
Assigns zeroes to large storage areas without actually sending the zeros to the storage system. This speeds up the VM initiation operation, and reduces the I/O and CPU load on the host.

Hardware-assisted locking
Locks a particular range of blocks in a shared logical unit, providing exclusive access to these blocks. Instead of using SCSI reservation that locks the entire logical unit, locking specific blocks is a more efficient alternative that greatly improves scalability in large server arrays. The locking is performed using Atomic Test & Set (ATS) commands.

The offloading of these extended operations from the ESXi server to the storage system reduces strain on the ESXi server, and saves a significant amount of time and computing resources when performing these operations in a full-scale production environment.

Integration with Microsoft Azure Site Recovery

The Microsoft Azure Site Recovery (ASR) solution helps protect important applications by coordinating the replication and recovery of private clouds across remote sites.

IBM FlashSystem A9000 and A9000R can be integrated with Microsoft Azure Site Recovery, enabling customers using Microsoft System Center Virtual Machine Manager (SCVMM) to orchestrate and manage replication and disaster recovery.

Support for Microsoft Azure Site Recovery is based on support for SMI-S (http://www.snia.org/ctp/conformingproviders/ibm.html#sftw7).

The SCVMM ASR integrates with storage solutions, such as the FlashSystem A9000 and A9000R CIM Agent, to provide site-to-site disaster recovery for Hyper-V environments by leveraging the SAN replication capabilities that are natively offered by the storage system. It orchestrates replication and failover for virtual machines managed by SCVMM.

SCVMM ASR uses the remote mirroring feature (see Chapter 11, “Synchronous remote mirroring,” on page 55) through SMI-S to create and manage the replication groups.

Chapter 23. Software upgrade

Non-disruptive code load (hot upgrade) allows upgrading the storage system software from a current version to a newer version without disrupting the storage provisioning service.

Important: Refer to the latest release notes for information on available upgrade paths for non-disruptive code load. Some software versions might not support non-disruptive code load, depending on the design of these software versions.

During upgrade process there is a point in time dubbed as the 'point-of-no-return', before which the process can still be aborted (either automatically by the system - or manually through a dedicated CLI). Once that point is crossed, the upgrade process is not reversible.

The upgrade process is run on all modules in parallel and is designed to be quick enough so that the applications' service on the hosts will not be damaged. The upgrade requires that neither data migration nor rebuild processes are run, and that all internal network paths are active.

Following are notable characteristics of the Non-disruptive code load:

Duration of the upgrade process

The overall process of downloading new code to storage system and moving to the new code is done online to the application/Host.

The duration of the upgrade process is affected by the following factors:

- The upgrade process requires that you reduce all I/Os. If there are a lot of I/Os in the system, or there are slow disks, the system might not be able to stop the I/Os fast enough, so it will restart them and try again after a short while, taking into consideration some retries.
- The upgrade process installs a valid version of the software and then retains its local configuration. This process might take a considerable amount of time, depending on the future changes in the structure of the configuration.

Prerequisites and constraints

- The process cannot run if a data migration process or a rebuild process is active. An attempt to start the upgrade process when either a data migration or a rebuild process is active will fail.
- Generally, everything that happens after the 'point-of-no-return' is treated as if it happened after the upgrade is over.
- As long as the overall hot upgrade is in progress (up to several minutes) no management operations are allowed (save for status querying), and no events are processed.
- Prior to the 'point-of-no-return', a manual abort of the upgrade is available.

Effect on mirroring

Mirrors are automatically deactivated before the upgrade, and reactivated after it is over.

Effect on management operations

During the upgrade process it is possible to query the system about the upgrade status, and the process can also be aborted manually before the 'point-of-no-return'. If a failure occurs before this point, the process is aborted automatically.

Preparing for software upgrade

The system code is upgraded without disconnecting active hosts or stopping I/O operations. However, some preparation is required.

Attention: The upgrade must be performed only by an authorized IBM service technician.

Before the code load (software upgrade), fulfill the following prerequisites by verifying that:

1. The native multipathing feature (provided by the operating system) is working on the connected hosts.
2. There are paths from the host to at least two different interface modules on the storage system.
3. There is no more than a single initiator in each zone (SAN Volume Controller attached to the system is an exception).
4. All hosts were attached to the storage system by using the IBM Storage Host Attachment Kit (HAK).
   - This applies to both the installable HAK and portable HAK versions.
   - Exceptions to this prerequisite are supported platforms, for which no HAK is available (for example, VMware hosts or Linux on Power hosts).
5. In a case the storage system uses Fibre Channel (FC) connectivity for remote mirroring, the two systems should be connected to a SAN switch. Direct connection is not supported and is known to be problematic.
6. Hosts should be attached to the FC ports through an FC switch, and to the iSCSI ports through a Gigabit Ethernet switch. Direct attachment between hosts and to the storage system is not supported.

In addition, the following must be taken into consideration:

1. Co-existence with other multipathing software is not supported as GA (RPQ approval is required).
2. Connectivity to other storage servers from the same host is not supported as GA (RPQ approval is required).
3. Any remote mirroring connection is automatically suspended and resumed for a short while during the upgrade.
4. If a Microsoft Geo Cluster is used, check the cluster requirements.

Chapter 24. Remote support and proactive support

To allow IBM to provide support for the storage system, the proactive support and remote support options are available.

Note: For various preventive and diagnostics support actions relating to the storage system's continuous operation, IBM Support requires customer approval. Without customer approval, these support actions cannot be preformed.

- **Proactive support** ("Call Home") – Allows proactive notifications regarding the storage system health and components to be sent to IBM Support at predefined intervals. Heartbeats and events are sent from the system to the IBM service center. The service center analyzes the information within the heartbeats and the events, correlates it with its vast database and can then trigger a component replacement prior to its potential failure.

  Upon detection of any hardware or software error code, both IBM Support and your predefined contact person are notified via e-mail, through a specified SMTP gateway. If IBM Support determines that the detected event requires service or further investigation, a new PMR is created and sent to appropriate IBM Support team. Proactive support minimizes the number of interaction cycles with IBM Support.

  If required, the customer email gateway can be configured to send call home information to IBM only via a secured channel. For more information, refer to the 'Encrypting Call Home and heartbeat notifications' section in the *IBM FlashSystem A9000 and IBM FlashSystem A9000R Architecture and Implementation* Redbooks publication.
- **Remote support** – Allows IBM Support to remotely and securely access your storage system when needed during a support call. This option requires IP communication between the storage system and the IBM Remote Support Center. If a storage system does not have direct access to the Internet (for example, due to a firewall), use the IBM Remote Support Proxy utility to enable the connection. Remote support minimizes the time it takes to diagnose and remedy storage system operational issues.

For more information, refer to the Planning for remote support, on-site service, and maintenance section of the *Deployment Guide* (ibm.com/support/knowledgecenter/STJKN5_12.2.1/fs_podrack_dg_planning_support.html).

Notices

These legal notices pertain to the information in this IBM Storage product documentation.

This information was developed for products and services offered in the US. This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Trademarks

IBM, IBM FlashSystem, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Copyright and trademark information website (ibm.com/legal/us/en/copytrade.shtml).

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Microsoft, Windows, Windows Server, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries, or both.

VMware, ESX, ESXi, vSphere, vCenter, and vCloud are trademarks or registered trademarks of VMware Corporation in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

Printed in USA

SC27-8558-11