

# Dell EMC Storage Systems

## Product Guide for PowerStore and Unity XT metro node feature

Version 7.0

## Notes, cautions, and warnings

 **NOTE:** A NOTE indicates important information that helps you make better use of your product.

 **CAUTION:** A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

 **WARNING:** A WARNING indicates a potential for property damage, personal injury, or death.

<b>Figures.....</b>	<b>5</b>
<b>Tables.....</b>	<b>6</b>
<b>Preface.....</b>	<b>7</b>
<b>Chapter 1: Introducing metro node.....</b>	<b>9</b>
Metro node overview.....	9
Metro node product family.....	10
Metro node Local.....	10
Metro node Metro.....	11
Metro node hardware platforms.....	11
Configuration highlights.....	11
Management interfaces.....	12
Web-based GUI.....	12
Metro node CLI.....	13
Metro node Element Manager API.....	13
<b>Chapter 2: Metro node use cases.....</b>	<b>14</b>
General use cases and benefits.....	14
Mobility.....	14
Technology refresh.....	16
Availability.....	17
<b>Chapter 3: Features in metro node.....</b>	<b>19</b>
Metro node security features.....	19
ALUA.....	19
Provisioning with metro node.....	20
Support for thin volumes and unmapping.....	20
Performance monitoring .....	21
Unisphere Performance Monitoring Dashboard.....	21
Performance monitoring using the CLI.....	22
Notification.....	22
<b>Chapter 4: Integrity and resiliency.....</b>	<b>23</b>
About metro node resilience and integrity.....	23
Site distribution.....	23
Cluster.....	24
Quorum.....	24
Metadata volumes.....	25
Backup metadata volumes.....	25
Logging volumes.....	25
High availability and metro node hardware.....	26
Directors.....	26
Management server.....	28


Metro node Metro Hardware.....	29
<b>Chapter 5: Software and upgrade.....</b>	<b>30</b>
Metro node OS.....	30
Non-disruptive upgrade (NDU).....	31
Storage, application, and host upgrades.....	31
Software upgrades.....	31
Simple support matrix.....	31
<b>Glossary.....</b>	<b>32</b>
<b>Index.....</b>	<b>43</b>

1	Metro node active-active.....	9
2	Metro node family: Local and Metro.....	10
3	Configuration highlights.....	12
4	Claim storage using the GUI (for HTML5).....	13
5	Moving data with metro node.....	15
6	Metro node technology refresh.....	17
7	High availability infrastructure example.....	18
8	Unisphere Performance Monitoring Dashboard (for HTML5).....	21
9	Unisphere Performance Monitoring Dashboard - select information to view (for HTML5).....	21
10	Unisphere Performance Monitoring Dashboard - sample chart (for UI).....	22
11	Path redundancy: different sites.....	24
12	Path redundancy: different ports.....	27
13	Path redundancy: different directors.....	28

1	Typographical conventions.....	8
2	General metro node use cases and benefits.....	14
3	Types of data mobility operations.....	15
4	Metro node OS AccessAnywhere features.....	30

As part of an effort to improve its product lines, Dell EMC periodically releases revisions of its software and hardware. Therefore, some functions described in this document might not be supported by all versions of the software or hardware currently in use. The product release notes provide the most up-to-date information on product features.

Contact your Dell EMC technical support professional if a product does not function properly or does not function as described in this document.

 **NOTE:** This document was accurate at publication time. Go to Dell EMC Online Support (<https://www.dell.com/support>) to ensure that you are using the latest version of this document.

## Purpose

This document is part of the VPLEX documentation set, and describes the VPLEX features and use cases, configuration options, VPLEX software and its upgrade, and the hardware overview.

## Audience

This guide is intended for use by customers who wish to understand the software and hardware features of VPLEX, the use cases of VPLEX, product offerings, and the configuration options.


Related documents (available on Dell EMC Online Support) include:


- *Release Notes for metro node*
- *Product Guide for metro node*
- *metro node Hardware Environment Setup Guide*
- *Configuration Guide for metro node*
- *Installation Guide for metro node*
- *Security Configuration Guide for metro node*
- *CLI Reference Guide for metro node*
- *Administration Guide for metro node*
- Online Help for metro node
- *Element Manager API Guide Version 2 (REST API v2) for metro node*
- *Open-Source Licenses Guide for metro node*
- Procedures provided through the SolVe Desktop
- *Dell EMC Host Connectivity Guides*
- *Hardware Reference Guide for metro node*
- Various best practices technical notes available on Dell EMC Online Support

## Special notice conventions used in this document


Dell EMC uses the following conventions for special notices:

 **CAUTION:** Indicates a hazardous situation which, if not avoided, will result in death or serious injury.

 **CAUTION:** Indicates a hazardous situation which, if not avoided, could result in death or serious injury.

 **CAUTION:** Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

 **NOTE:** Addresses practices not related to personal injury.

 **NOTE:** Presents information that is important, but not hazard-related.

## Typographical conventions

Dell EMC uses the following type style conventions in this document:

**Table 1. Typographical conventions**

<b>Bold</b>	Used for names of interface elements, such as names of windows, dialog boxes, buttons, fields, tab names, key names, and menu paths (what the user specifically selects or clicks)
<i>italic</i>	Used for full titles of publications referenced in text
Monospace	Used for: <ul style="list-style-type: none"><li>• System code</li><li>• System output, such as an error message or script</li><li>• Pathnames, filenames, prompts, and syntax</li><li>• Commands and options</li></ul>
<i>Monospace italic</i>	Used for variables
<b>Monospace bold</b>	Used for user input
[ ]	Square brackets enclose optional values
	Vertical bar indicates alternate selections - the bar means "or"
{ }	Braces enclose content that the user must specify, such as x or y or z
...	Ellipses indicate nonessential information omitted from the example

## Where to get help

Dell EMC support, product, and licensing information can be obtained as follows:

## Product information

For documentation, release notes, software updates, or information about Dell EMC products, go to Dell EMC Online Support at <https://www.dell.com/support>.

## Technical support

Go to Dell EMC Online Support and click Support. You will see several options for contacting Dell EMC Technical Support. Note that to open a service request, you must have a valid support agreement. Contact your Dell EMC sales representative for details about obtaining a valid support agreement or with questions about your account.

## Online communities

Visit Dell EMC Community Network (DECN) at <https://www.dell.com/community/Dell-Community/ct-p/English> for peer contacts, conversations, and content on product support and solutions. Interactively engage online with customers, partners, and certified professionals for all Dell EMC products.

## Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Send your opinions of this document to [vplex.doc.feedback@dell.com](mailto:vplex.doc.feedback@dell.com).



# Introducing metro node

This chapter introduces the metro node feature.

## Topics:

- [Metro node overview](#)
- [Metro node product family](#)
- [Metro node hardware platforms](#)
- [Configuration highlights](#)
- [Management interfaces](#)

## Metro node overview

Metro node virtualizes data that is on storage arrays to create dynamic, distributed, and highly available data centers.

Use metro node to:

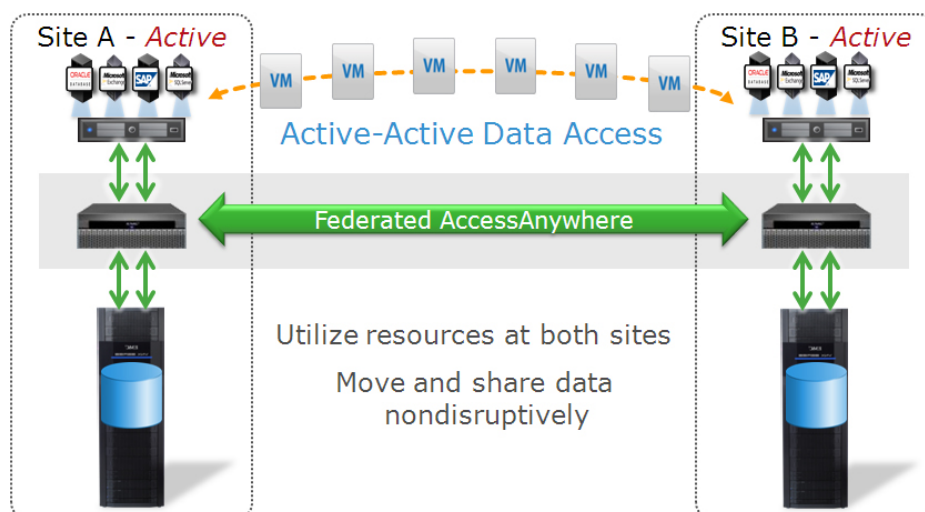
- Move data nondisruptively between Dell EMC PowerStore, Unity XT and other third-party storage arrays without any downtime for the host.

Metro node moves data transparently, and the virtual volumes retain the same identities and the same access points to the host. There is no need to reconfigure the host.

- Protect data in the event of disasters or failure of components in your data centers.

With metro node, you can withstand failures of storage arrays, cluster components, an entire site failure, or loss of communication between sites (when two clusters are deployed) and still keep applications and data online and available.

With metro node, you can transform the delivery of IT to a flexible, efficient, reliable, and resilient service.



**Figure 1. Metro node active-active**

Metro node addresses these two primary IT needs:

- **Mobility:** Metro node moves applications and data between different storage installations:
  - Within the same data center or across a campus (metro node Local)
  - Within a geographical region (metro node Metro)
- **Availability:** Metro node creates high-availability storage infrastructure across these same varied geographies with unmatched resiliency.

Metro node offers the following unique innovations and advantages:

- Metro node distributed/federated virtual storage enables new models of application and Data Mobility.  
Metro node is optimized for virtual server platforms (VMware ESX, Hyper-V, Oracle Virtual Machine, AIX VIOS).  
Metro node can streamline or accelerate transparent workload relocation over distances, including moving virtual machines.
- In a Metro configuration, metro node AccessAnywhere provides image consistent active-active access to data across two metro node clusters.

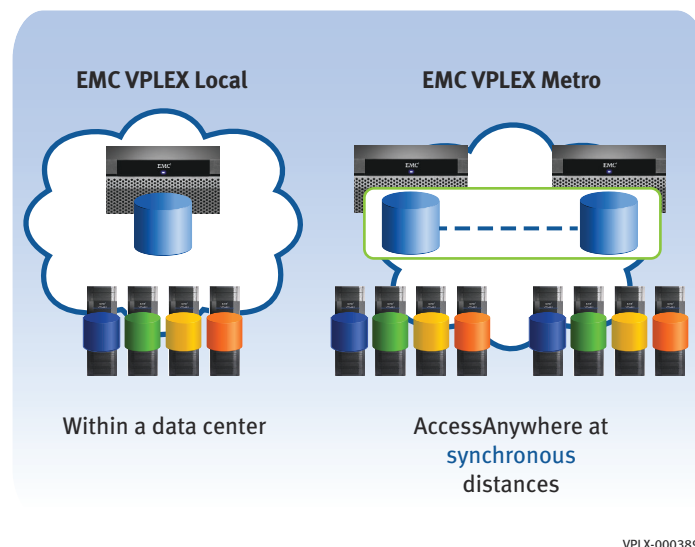
Metro node pools the storage resources in multiple data centers so that the data can be accessed anywhere. With metro node, you can:

- Provide continuous availability and workload mobility.
- Replace your tedious data movement and technology refresh processes with metro node's patented simple, frictionless two-way data exchange between locations.
- Create an active-active configuration for the active use of resources at both sites.
- Provide instant access to data between data centers. metro node allows simple, frictionless two-way data exchange between locations.
- Combine metro node with virtual servers to enable private and hybrid cloud computing.

## Metro node product family

The metro node product family includes:

- Metro node Local
- Metro node Metro



**Figure 2. Metro node family: Local and Metro**

## Metro node Local

Metro node Local consists of a single cluster. metro node Local:

- Federates Dell EMC PowerStore and Unity XT arrays with other Dell EMC and non-Dell EMC storage arrays.  
Federation allows transparent data mobility between arrays for simple, fast data movement and technology refreshes.
- Standardizes LUN presentation and management using simple tools to provision and allocate virtualized storage devices.
- Improves storage utilization using pooling and capacity aggregation across multiple arrays.
- Increases protection and high availability for critical applications.  
Mirrors storage across mixed platforms without host resources.  
Leverage your existing storage resources to deliver increased protection and availability for critical applications.

Deploy metro node Local within a single data center.

## Metro node Metro

Metro node Metro consists of two metro node clusters connected by inter-cluster links with not more than 10ms Round Trip Time (RTT). metro node Metro:

- Transparently relocates data and applications over distance, protects your data center against disaster.  
Manage all of your storage in both data centers from one management interface.
- Mirrors your data to a second site, with full access at near local speeds.

Deploy metro node Metro within a data center for:

- Additional virtual storage capabilities beyond that of a metro node Local.
- Higher availability.

Metro clusters can be placed up to 100 km apart, allowing them to be located at opposite ends of an equipment room, on different floors, or in different fire suppression zones; all of which might be the difference between riding through a local fault or fire without an outage.

Deploy metro node Metro between data centers for:

- Mobility: Redistribute application workloads between the two data centers.
- Availability: Applications must keep running in the presence of data center failures.
- Distribution: One data center lacks space, power, or cooling.

Combine metro node Metro virtual storage and virtual servers to:

- Transparently move virtual machines and storage across synchronous distances.
- Improve utilization and availability across heterogeneous arrays and multiple sites.

Distance between clusters is limited by physical distance, by host, and by application requirements. metro node Metro clusters contain additional I/O modules to enable the inter-cluster WAN communication over IP or Fibre Channel.

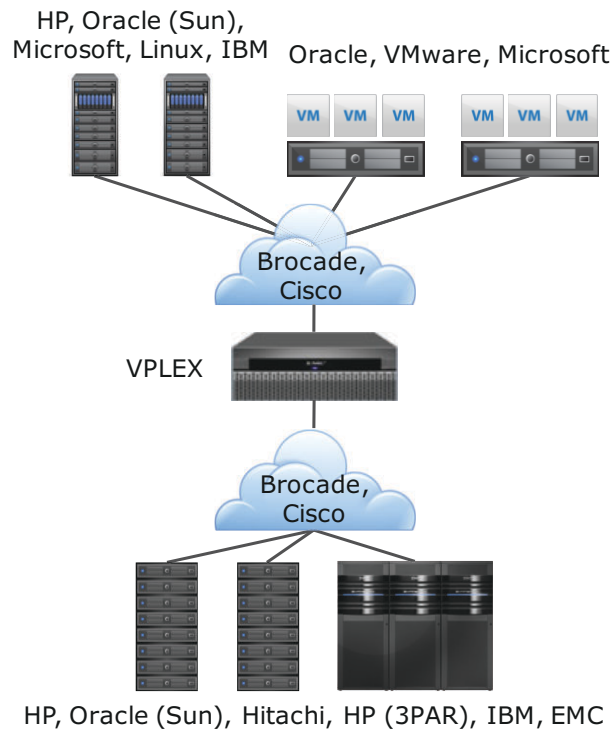
## Metro node hardware platforms

Metro node hardware platform is based on the Dell R640 PowerEdge server.

## Configuration highlights

A metro node primarily consists of:

- Two hardware nodes
- The two nodes are directly and redundantly connected using two Cat6 shielded cables for management connectivity and two Dell Direct-Attach-Copper (DAC) cables with SFP plugs for local data com connectivity.
- Two 32 Gig, 2 port FC HBA for FE and BE connectivity.
- The management server runs virtually on the metro node hardware . Each node has a public Ethernet port which provides cluster management services when connected to the network.



**Figure 3. Configuration highlights**

Metro node conforms to established world wide naming (WWN) guidelines that can be used for zoning. It also supports Dell EMC storage and arrays from other storage vendors, such as HDS, HP, and IBM. metro node provides storage federation for operating systems and applications that support clustered file systems, including both physical and virtual server environments with VMware, ESX, and Microsoft Hyper-V. The network fabrics from Brocade and Cisco are supported in metro node.

See the *Dell EMC Simple Support Matrix*, available at <http://elabnavigator.EMC.com> under the Simple Support Matrix tab.

## Management interfaces

In a metro node Metro configuration, both clusters can be managed from either management server.

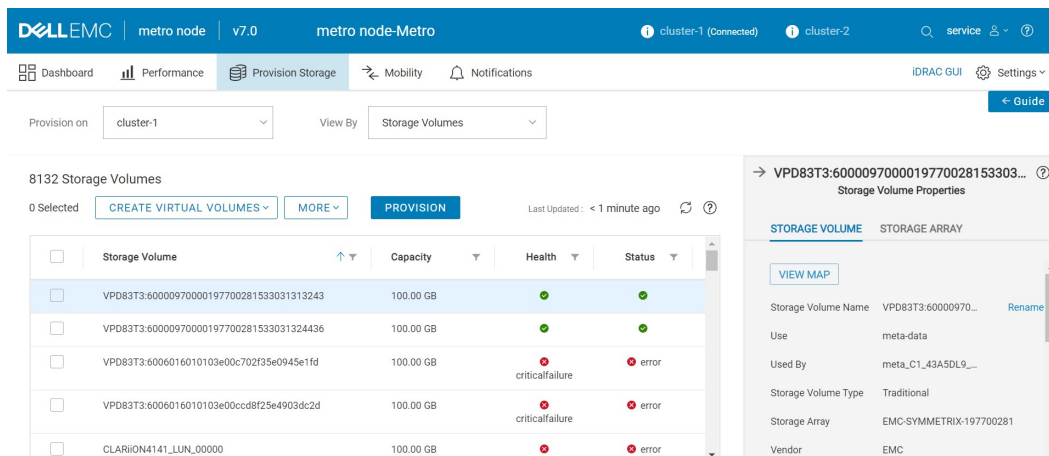
Inside metro node clusters, management traffic traverses a TCP/IP based private management network.

In a metro node Metro configuration, the management traffic between the clusters is secured through https protocol.

## Web-based GUI

Metro node web-based user interface (UI) provides an easy-to-use point-and-click management interface.

The following figures show the screen to claim storage:



**Figure 4. Claim storage using the GUI (for HTML5)**

The UI supports most of the metro node operations, and includes Dell EMC Online help for metro node to assist new users in learning the interface.

Metro node operations that are not available in the GUI, are supported by the Command Line Interface (CLI), which supports full functionality.

## Metro node CLI

The metro node CLI supports all metro node operations.

The CLI is divided into command contexts:

- Global commands are accessible from all contexts.
- Other commands are arranged in a hierarchical context tree, and can be executed only from the appropriate location in the context tree.

The following example shows a CLI session that performs the same tasks as shown in [Figure](#).

### Example 1 Claim storage using the CLI:

In the following example, the `claimingwizard` command finds unclaimed storage volumes, claims them as thin storage, and assigns names from a CLARiiON hints file:

```
Vplexcli:/clusters/cluster-1/storage-elements/
storage-volumes> claimingwizard --file /home/service/clar.txt
--thin-rebuild
Found unclaimed storage-volume
VPD83T3:6006016091c50e004f57534d0c17e011 vendor DGC:
claiming and naming clar_LUN82.
Found unclaimed storage-volume
VPD83T3:6006016091c50e005157534d0c17e011 vendor DGC:
claiming and naming clar_LUN84.
Claimed 2 storage-volumes in storage array car
Claimed 2 storage-volumes in total.
Vplexcli:/clusters/cluster-1/storage-elements/storage-volumes>
```

The *Dell EMC CLI Guide for metro node* provides a comprehensive list of metro node commands and detailed instructions on using those commands.

## Metro node Element Manager API

Metro node Element Manager API uses the Representational State Transfer (REST) software architecture for distributed systems such as the World Wide Web. It allows software developers and other users to use the API to create scripts to run metro node CLI commands.

Metro node Element Manager API supports all metro node CLI commands that can be run from the root context.

## Metro node use cases

This chapter describes the general features, benefits, and the important use cases of metro node.

### Topics:

- [General use cases and benefits](#)
- [Mobility](#)
- [Availability](#)

## General use cases and benefits

The following table summarizes the general metro node use cases and their benefits.

**Table 2. General metro node use cases and benefits**

General use cases	Benefits
Mobility	<ul style="list-style-type: none"> <li>• <b>Migration:</b> Move data and applications without impact on users.</li> <li>• <b>Virtual Storage federation:</b> Achieve transparent mobility and access within a data center and between data centers.</li> <li>• <b>Scale-out cluster architecture:</b> Start small and grow larger with predictable service levels.</li> </ul>
Availability	<ul style="list-style-type: none"> <li>• <b>Resiliency:</b> Mirror across arrays within a single data center or between data centers without host impact. This increases availability for critical applications.</li> <li>• <b>Distributed cache coherency:</b> Automate sharing, balancing, and failover of I/O across the cluster and between clusters whenever possible.</li> <li>• <b>Advanced data caching:</b> Improve I/O performance and reduce storage array contention.</li> </ul>

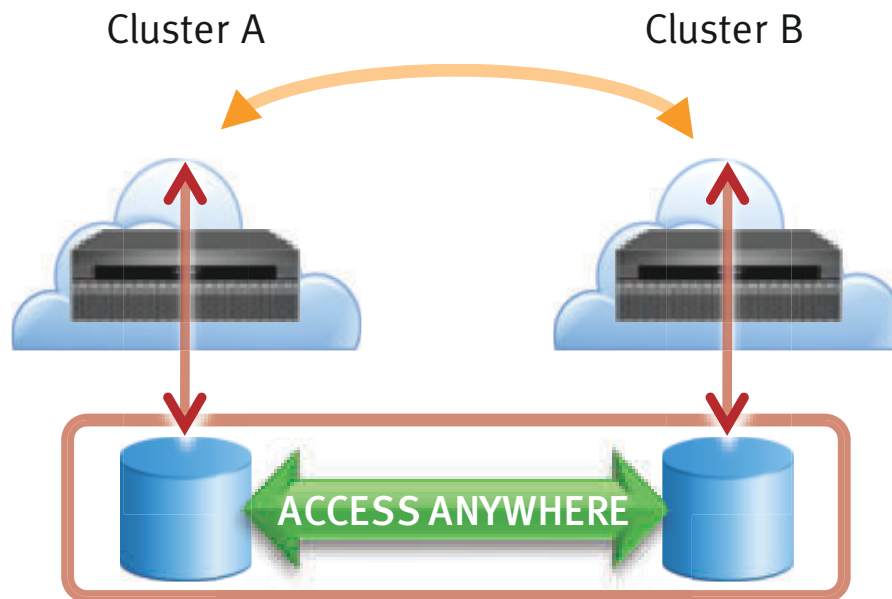
For all metro node deployments, do the following:

- Presents storage volumes from back-end arrays to metro node engines.
- Federates the storage volumes into hierarchies of metro node virtual volumes with user-defined configuration and protection levels.
- Presents virtual volumes to production hosts in the SAN through the metro node front-end.
- For metro node Metro, presents a global, block-level directory for distributed cache and I/O between metro node clusters.

## Mobility

Use metro node to move data between data centers, relocate a data center or consolidate data, without disrupting host application access to the data.

# MOBILITY



## Move and relocate VMs, application, and data over distance

**Figure 5. Moving data with metro node**

The source and target arrays can be in the same data center (metro node Local) or in different data centers separated by up to 10ms (metro node Metro). The source and target arrays can be heterogeneous.

When you use metro node to move data, the data retains its original metro node volume identifier during and after the mobility operation. No change in volume identifiers eliminates application cut over. The application continues to use the same data, though the data has been moved to a different storage array.

There are many types and reasons to move data:

- Move data from a hot storage device.
- Move the data from one storage device to another without moving the application.
- Move operating system files from one storage device to another.
- Consolidate data or database instances.
- Move database instances.
- Move storage infrastructure from one physical location to another.

With metro node, you no longer need to spend significant time and resources preparing to move data and applications. You do not have to plan for an application downtime or restart the applications as part of the data movement activity. Instead, a move can be made instantly between sites, over distance, and the data remains online and available during the move without any outage or downtime. Considerations before moving the data include the business impact, type of data to be moved, site locations, total amount of data, and schedules.

The data mobility feature of metro node is useful for disaster avoidance, planned upgrade, or physical movement of facilities. The mobility jobs in metro node are as follows:

**Table 3. Types of data mobility operations**

Device	Moves data from one device to another device (within a cluster and across clusters).
--------	--

**Table 3. Types of data mobility operations (continued)**

Batch	<p>Moves data using a migration plan file. Create batch migrations to automate routine tasks.</p> <ul style="list-style-type: none"><li>• Use batched device migrations to migrate to dissimilar arrays and to migrate devices within a cluster and between the clusters in a metro node Metro configuration.</li></ul>
-------	---

## Technology refresh

In typical IT environments, migrations to new storage arrays (technology refreshes) require that the data that is being used by hosts be copied to a new volume on the new array. The host must then be reconfigured to access the new storage. This process requires downtime for the host.

Metro node makes it easier to replace heterogeneous storage arrays on the back-end. Migrations between heterogeneous arrays can be complicated and may require additional software or functionality. Integrating heterogeneous arrays in a single environment is difficult and requires a staff with a diverse skill set.

When metro node is inserted between the front-end and back-end redundant fabrics, metro node appears as the target to hosts and as the initiator to storage.

The data resides on virtual volumes in metro node and it can be copied nondisruptively from one array to another without any downtime. There is no need to reconfigure the host; the physical data relocation is performed by metro node transparently and the virtual volumes retain the same identities and the same access points to the host.

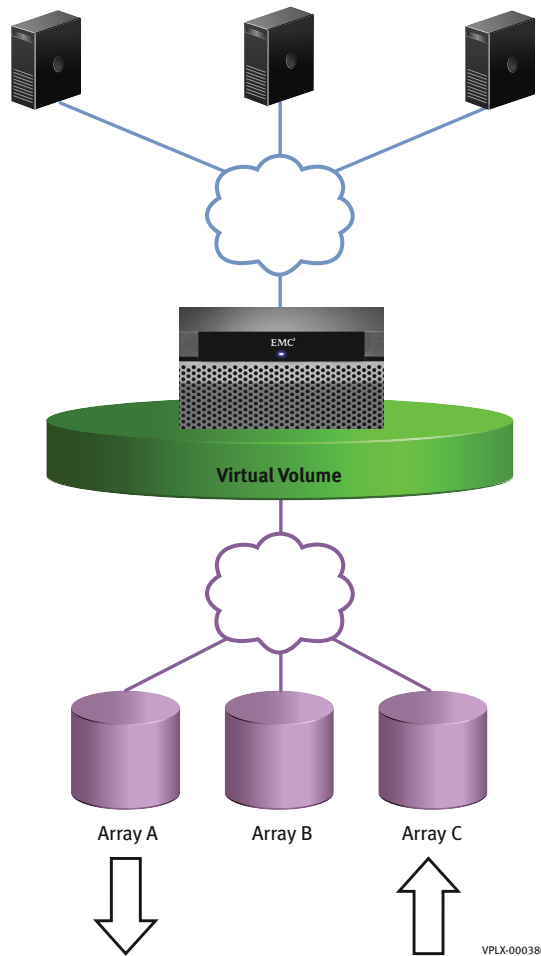
In the following figure, the virtual disk is made up of the disks of Array A and Array B. The site administrator has determined that Array A has become obsolete and should be replaced with a new array. Array C is the new storage array. Using Mobility Central, the administrator:

- Adds Array C into the metro node cluster.
- Assigns a target extent from the new array to each extent from the old array.
- Instructs metro node to perform the migration.

Metro node copies data from Array A to Array C while the host continues its access to the virtual volume without disruption.

After the copy of Array A to Array C is complete, Array A can be decommissioned:





**Figure 6. Metro node technology refresh**

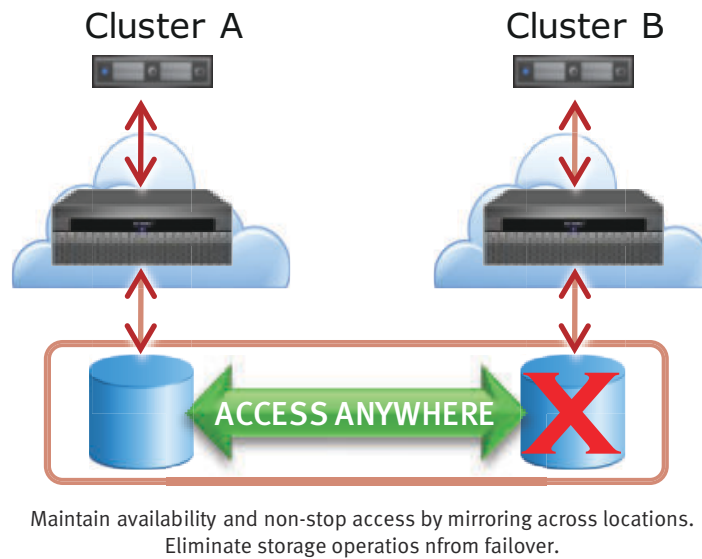
Because the virtual machine is addressing its data to the abstracted virtual volume, its data continues to flow to the virtual volume without any need to change the address of the data store.

Although this example uses virtual machines, the same is true for traditional hosts. Using metro node, the administrator can move data used by an application to a different storage array without the application or server being aware of the change.

This allows you to change the back-end storage arrays transparently, without interrupting I/O.

## Availability

Metro node features allow the highest possible resiliency in the event of an outage. The following figure shows a metro node Metro configuration where storage has become unavailable at one of the cluster sites.



**Figure 7. High availability infrastructure example**

Metro node redundancy provides reduced Recovery Time Objective (RTO) and Recovery Point Objective (RPO). Because metro node AccessAnywhere mirrors all data, applications continue without disruption using the back-end storage at the unaffected site.

With the Federated AccessAnywhere feature of metro node, the data remains consistent, online, and always available. metro node does not need to ship the entire file back and forth like other solutions. It only sends the changed updates as they are made, greatly reducing bandwidth costs and offering significant savings over other solutions.

To know more about high availability with metro node, see [Chapter 4 Integrity and resiliency](#).

# Features in metro node

This chapter describes the specific features of metro node.

## Topics:

- [Metro node security features](#)
- [ALUA](#)
- [Provisioning with metro node](#)
- [Performance monitoring](#)
- [Notification](#)


## Metro node security features

The operating systems of the metro node management server and the directors are based on a Novell SUSE Linux Enterprise Server 15 SP1 distribution.

The operating system has been configured to meet Dell EMC security standards by disabling or removing unused services, and protecting access to network services through a firewall.

The metro node security features include:

- LDAP authentication using OS secure SSSD service
- HTTPS to access the metro node UI
- HTTPS inter-cluster link in a metro node Metro configuration
- SCP to copy files
- Support for separate networks for all metro node cluster communication
- Defined user accounts and roles
- Certificate Authority (CA) certificate (default expiration 5 years)
- Two host certificates (default expiration 2 years)
- External directory server support

 **CAUTION:** The WAN-COM inter-cluster link carries unencrypted user data. To ensure privacy of the data, establish an encrypted VPN tunnel between the two sites.

For more information about security features and configuration see the *Dell EMC Security Configuration Guide for metro node*.

## ALUA

Asymmetric Logical Unit Access (ALUA) routes I/O of the LUN directed to non-active/failed storage processor to the active storage processor without changing the ownership of the LUN.

Each LUN has two types of paths:

- **Active/optimized paths** are direct paths to the storage processor that owns the LUN.  
Active/optimized paths are usually the optimal path and provide higher bandwidth than active/non-optimized paths.
- **Active/non-optimized paths** are indirect paths to the storage processor that does not own the LUN through an interconnect bus.  
I/Os that traverse through the active/non-optimized paths must be transferred to the storage processor that owns the LUN. This transfer increases latency and has an impact on the array.

Metro node detects the different path types and performs round robin load balancing across the active/optimized paths.

Metro node supports all three flavors of ALUA:

- **Explicit ALUA** - The storage processor changes the state of paths in response to commands (for example, the Set Target Port Groups command) from the host (the metro node backend).

The storage processor must be explicitly instructed to change a path's state.

If the active/optimized path fails, metro node issues the instruction to transition the active/non-optimized path to active/optimized.

There is no need to failover the LUN.

- **Implicit ALUA** - The storage processor can change the state of a path without any command from the host (the metro node back end).

If the controller that owns the LUN fails, the array changes the state of the active/non-optimized path to active/optimized and fails over the LUN from the failed controller.

On the next I/O, after changing the path's state, the storage processor returns a Unit Attention "Asymmetric Access State Changed" to the host (the metro node backend).

Metro node then re-discovers all the paths to get the updated access states.

- **Implicit/explicit ALUA** - Either the host or the array can initiate the access state change.

Storage processors support implicit only, explicit only, or both.

## Provisioning with metro node

Metro node allows easy storage provisioning among heterogeneous storage arrays. Use the web-based GUI to simplify everyday provisioning or create complex devices.

There are two ways to provision storage in metro node:

- EZ provisioning
- Advanced provisioning

All provisioning features are available in the Unisphere for metro node UI.

## Support for thin volumes and unmapping

Thin provisioning advertises the metro node virtual volumes as thin volumes to the hosts. Thin provisioning dynamically allocates block resources only when they are required. It essentially allows efficient utilization of physical block resources from the storage arrays.

Hosts gather the properties related to the thin provisioning feature of a metro node virtual volume and send SCSI commands to free storage block resources that are not in use. If the blocks of the back end storage volumes are free, the blocks can be mapped to other changed regions. Thin provisioning enables dynamic freeing of storage blocks on storage volumes for which thin provisioning is supported.

 **NOTE:** The *Dell EMC Simplified Support Matrix for metro node* provides more information on the supported storage volumes.

Metro node thin provisioning support includes the following features:

- Discovery of the back-end storage volumes capable for thin provisioning - During the back-end storage volume discovery, metro node gathers all thin provisioning related storage volume properties. metro node also performs a consistency check on all the properties related to thin-provisioning.
- Reporting thin provisioning enabled metro node virtual volumes to hosts - metro node shares the details of the thin provisioning-enabled virtual volumes with the hosts.
- Reclaiming the unused storage blocks - Through a command, metro node removes the mapping between a deleted virtual machine and its storage volumes and reclaims the storage blocks corresponding to the VMFS blocks used by that virtual machine.
- Handling storage exhaustion - The exhaustion of storage blocks on non-mirrored storage volumes are notified to the host as a space allocation failure. This error notification is posted to the host and the VMware hosts stop the impacted virtual machine.

To prevent potential mapping of all the blocks in the storage volumes that are thin capable, metro node uses thin rebuilds. Thin rebuilds can be configured to be set or unset for any claimed storage volume on which metro node builds virtual volumes. This property controls how metro node does its mirror rebuilding.

The unmap feature reclaims the unused VMFS blocks by removing the mapping between the logical blocks and the physical blocks. This essentially removes the link between a logical block and a physical block that has unknown or unused resources.

## Performance monitoring

Metro node performance monitoring provides a customized view into the performance of your system. You decide which aspects of the system's performance to view and compare.

You can view and assess the metro node performance using these methods:

- Unisphere Performance Monitoring Dashboard, which shows real-time performance monitoring data for up to one hour of history.
- Performance statistics collection using the CLI and the API. These methods let you collect and view the statistics, and export them to an external application for analysis.

## Unisphere Performance Monitoring Dashboard

The Unisphere Performance Monitoring Dashboard supports these general categories of performance monitoring:

- Current load monitoring that allows administrators to watch CPU load during upgrades, I/O load across the inter-cluster WAN link, and front-end against the back-end load during data mining or back up.
- Long term load monitoring that collects data for capacity planning and load balancing.
- Object-based monitoring that collects data for the virtual volume.

The Unisphere Performance Monitoring Dashboard is a customized view into the performance of the metro node system:

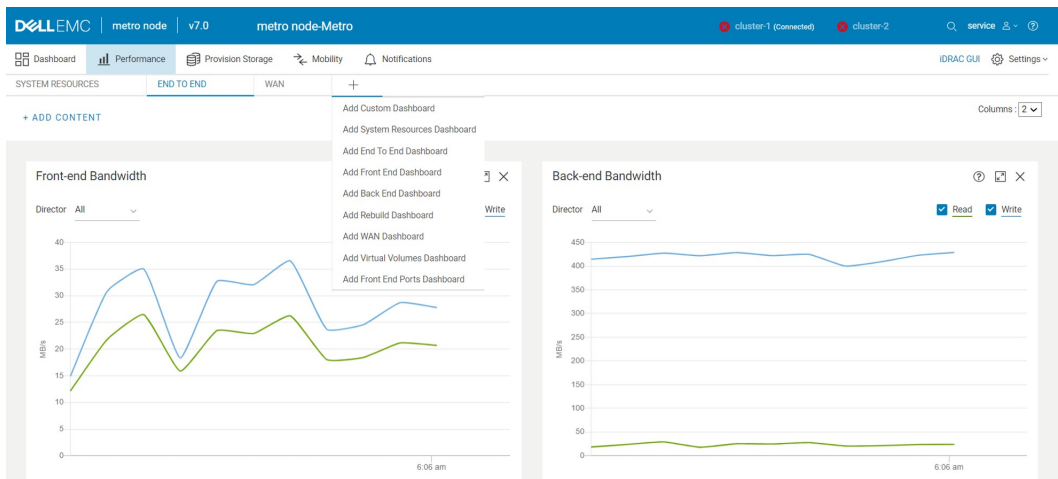


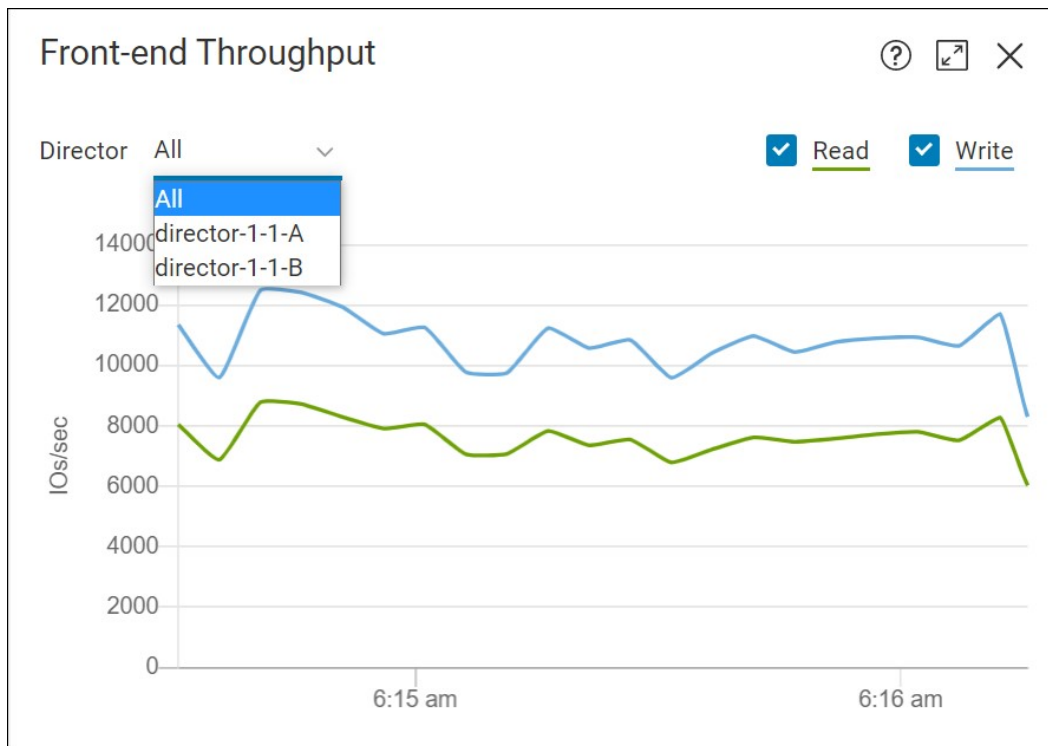
Figure 8. Unisphere Performance Monitoring Dashboard (for HTML5)

You decide which aspects of the system's performance to view and compare:



Figure 9. Unisphere Performance Monitoring Dashboard - select information to view (for HTML5)

Performance information is displayed as a set of charts. For example, the following figure shows front-end throughput for a selected director (for Flash) and all directors (for HTML5):



**Figure 10. Unisphere Performance Monitoring Dashboard - sample chart (for UI)**

For additional information about the statistics available through the Performance Monitoring Dashboard, see the Dell EMC Unisphere for metro node online help available in the metro node UI.

## Performance monitoring using the CLI

The CLI supports current load monitoring, long term load monitoring, object base monitoring, and troubleshooting monitoring. The CLI collects and displays performance statistics using:

**monitors** - Gather the specified statistic from the specified target at the specified interval.

**monitor sinks** - Direct the output to the desired destination. Monitor sinks include the console, a file, or a combination of the two.

Use two pre-defined perpetual monitors for each director to collect information to diagnose common problems.

Use the CLI to create a toolbox of custom monitors to operate under varying conditions including debugging, capacity planning, and workload characterization. For example:

The *Dell EMC Administration Guide for metro node* describes the procedure for monitoring metro node performance using the CLI.

## Notification

Events provide information about changes happening to the system, which also indicates that there is a problem with the system. Alerts are events that require attention by the system administrator or user. Most alerts indicate that there is a problem with the system that must be rectified to attain the best performance from the system.

The metro node notifications system displays live and historical alerts for platform, hardware (Both iDRAC and metro node Monitor alerts) in the Notification pane that requires user attention and helps to monitor the state of the various components, triage, and troubleshooting issues.

Notification features also allows to send alert notifications to a specified email or SMTP server. To configure SMTP server, see *System Configuration guide*.

# Integrity and resiliency

This chapter describes how the high availability and the redundancy features of metro node provide robust system integrity and resiliency.

## Topics:

- [About metro node resilience and integrity](#)
- [Site distribution](#)
- [Cluster](#)
- [Metadata volumes](#)
- [Backup metadata volumes](#)
- [Logging volumes](#)
- [High availability and metro node hardware](#)
- [Metro node Metro Hardware](#)

## About metro node resilience and integrity

With metro node, you get true high availability. Operations continue and data remains online even when a failure occurs. Within synchronous distances (metro node Metro), think of metro node as providing disaster avoidance instead of just disaster recovery.


Metro node Metro provides shared data access between sites. The same data (not a copy), exists at more than one location simultaneously. metro node can withstand a component failure, a site failure, or loss of communication between sites and still keep the application and data online and available. Metro node clusters are capable of surviving any single hardware failure in any subsystem within the overall storage cluster, including host connectivity and memory subsystems. A single failure in any subsystem does not affect the availability or integrity of the data.

Metro node redundancy creates fault tolerance for devices and hardware components that continue operation as long as one device or component survives. This highly available and robust architecture can sustain multiple device and component failures without disrupting service to I/O.

Failures and events that do not disrupt I/O include:

- Unplanned and planned storage outages
- SAN outages
- Metro node component failures
- Metro node cluster failures
- Data center outages

To achieve high availability, you must create redundant host connections and supply hosts with multi path drivers.

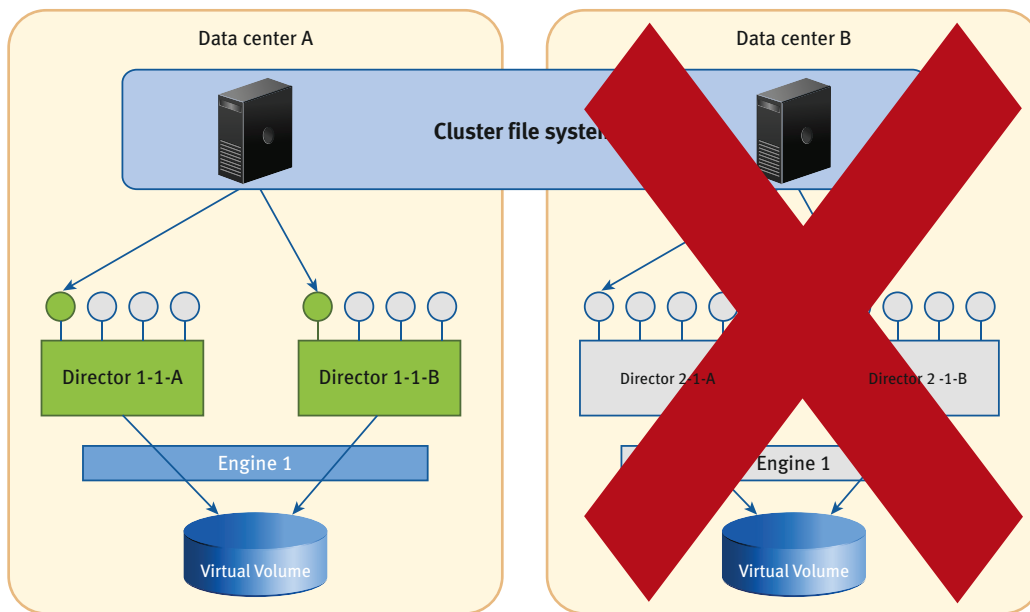
 **NOTE:** In the event of a front-end port failure or a director failure, hosts without redundant physical connectivity to a metro node cluster and without multi-pathing software installed could be susceptible to data unavailability.

## Site distribution

When two metro node clusters are connected together with metro node Metro, metro node gives you shared data access between sites. Metro node can withstand a component failure, a site failure, or loss of communication between sites and still keep the application and data online and available.

Metro node Metro ensures that if a data center goes down, or even if the link to that data center goes down, the other site can continue processing the host I/O.

In the following figure, despite a site failure at Data Center B, I/O continues without disruption in Data Center A.



VPLX-000394

Figure 11. Path redundancy: different sites

## Cluster

A metro node is a true cluster architecture. That is, all components are always available and I/O that enters the cluster from anywhere can be serviced by any node within the cluster, while cache and coherency is maintained for all reads and writes.

As you add more directors to the nodes, you get the added benefits of more cache, increased processing power, and more performance.

A metro node cluster provides N-1 fault tolerance, which means that any component failure can be sustained, and the cluster will continue to operate as long as one director survives.

A metro node cluster consists of redundant hardware components.

All hardware resources (CPU cycles, I/O ports, and cache memory) are pooled.

A two-cluster configuration (Metro) offers true high availability. Operations continue and data remains online even if an entire site fails. It also provides a high availability solution with zero recovery point objective (RPO).

## Quorum

Quorum refers to the minimum number of directors required for the cluster to service and maintain operations.

There are different quorum rules for a cluster to become operational and start servicing I/Os when it is booting up, also called "gaining quorum." Different rules for an operational cluster seeing director failures to either continue servicing operations and I/O after failure handling is called "maintaining quorum." Stopping servicing operations and I/O is called "losing quorum." These rules are described below:

- **Gaining quorum** - A non-operational metro node cluster gains quorum and becomes operational when more than half of the configured directors restart and come in contact with each other. In a single engine cluster, it refers to all the directors.
- **Maintaining quorum** - An operational metro node cluster seeing failures will continue operating in the following scenarios:
  - Director failures
    - If less than half of the operational directors with quorum fail.
    - If half of the operational directors with quorum fail, then the remaining directors will check the operational status of the failed directors over the management network and remain alive.



After recovering from this failure, a cluster can tolerate further similar director failures until only one director is remaining. In a single engine cluster, a maximum of one director failure can be tolerated.

- Intra-cluster communication failure
  - If there is a split in the middle, that is, half of the operational directors with quorum lose communication with the other half of the directors, and both halves are running, then the directors detect the operational status over the management network and instruct half with the director with the lowest UUID to keep running and the directors without the lowest UUID to operationally stop.
- **Quorum loss** - An operational metro node cluster seeing failures stops operating in the following scenarios:
  - If more than half of the operational directors with quorum fail at the same time.
  - If half of the operational directors with quorum fail, and the directors are unable to determine the operation status of the other half of the directors (whose membership includes a low UUID).
  - In a dual or quad engine cluster, if all of the directors lose contact with each other.

## Metadata volumes

Meta-volumes store metro node metadata, including virtual-to-physical mappings, data about devices, virtual volumes, and system configuration settings.

Metadata is stored in cache and backed up on specially designated external volumes called meta-volumes.

After the meta-volume is configured, updates to the metadata are written to both the cache and the meta-volume when the metro node configuration is modified.

Each metro node cluster maintains its own metadata, including:

- The local configuration for the cluster.
- Distributed configuration information shared between clusters.

At system startup, metro node reads the metadata and loads the configuration information onto each director.

When you make changes to the system configuration, metro node writes these changes to the metadata volume.

If metro node loses access to the metadata volume, the metro node directors continue uninterrupted, using the in-memory copy of the configuration. Metro node blocks changes to the system until access is restored or the automatic backup meta-volume is activated.

Meta-volumes experience high I/O only during system startup and upgrade.

I/O activity during normal operations is minimal.

## Backup metadata volumes

Backup metadata volumes are point-in-time snapshots of the current metadata, and provide extra protection before major configuration changes, refreshes, or migrations.

Backup creates a point-in-time copy of the current in-memory metadata without activating it. You must create a backup metadata volume in any of these conditions:

- As part of an overall system health check before a major migration or update.
- If metro node permanently loses access to active meta-volumes.
- After any major migration or update.


## Logging volumes

Logging volumes keep track of blocks written:

- During an inter-cluster link outage.
- When one leg of a DR1 becomes unreachable and then recovers.

After the inter-cluster link or leg is restored, the metro node system uses the information in logging volumes to synchronize the mirrors by sending only changed blocks across the link.

Logging volumes also track changes during loss of a volume when that volume is one mirror in a distributed device.

 **CAUTION:** If no logging volume is accessible, then the entire leg is marked as out-of-date. A full re-synchronization is required once the leg is reattached.

The logging volumes on the continuing cluster experience high I/O during:

- Network outages or cluster failures
- Incremental synchronization

When the network or cluster is restored, metro node reads the logging volume to determine what writes to synchronize to the reattached volume.

There is no I/O activity during normal operations.

## High availability and metro node hardware

The architectural design of the metro node hardware environment supports high availability.

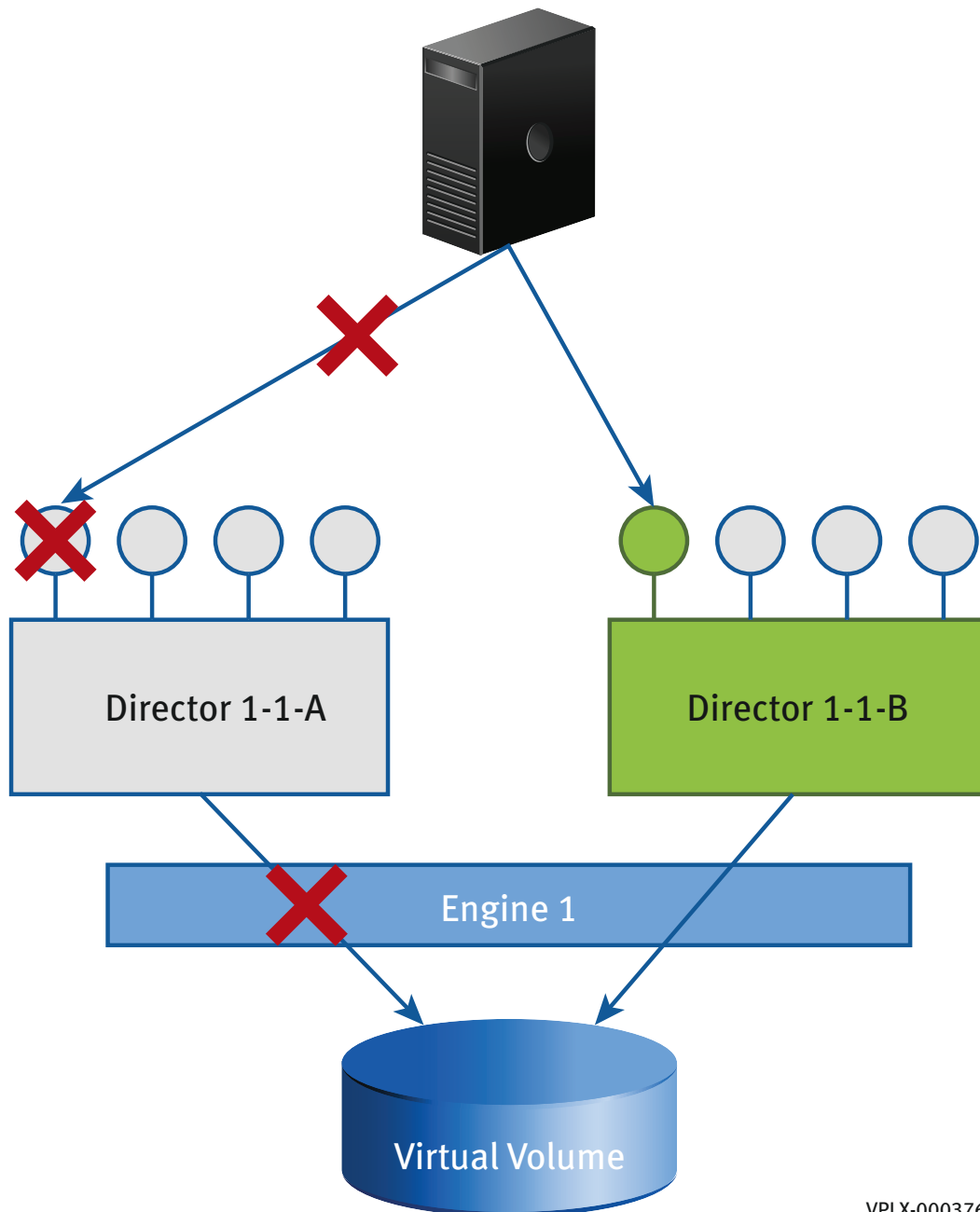
The metro node hardware is largely designed to withstand technical failures and provide uninterrupted data availability. The critical components in the hardware are redundant to ensure that the failure of a component does not bring the system down.

### Directors

A metro node director is the component that process the I/O requests from the hosts in a metro node environment. It interacts with the backend storage arrays for servicing the I/Os.

A director has two I/O modules for servicing I/Os from the arrays; one for the connectivity with the storage arrays on the back end, and another for connecting to the hosts on the front end. The management module in the director is used for management connectivity to the directors and for intra-cluster communication. The local communication module is completely dedicated to intra-cluster communication.

The front-end ports on all directors can provide access to any virtual volume in the cluster. Include multiple front-end ports in each storage view to protect against port failures. When a director port fails, the host multi-pathing software seamlessly fails over to another path through a different port, as shown in the following figure:



VPLX-000376

**Figure 12. Path redundancy: different ports**

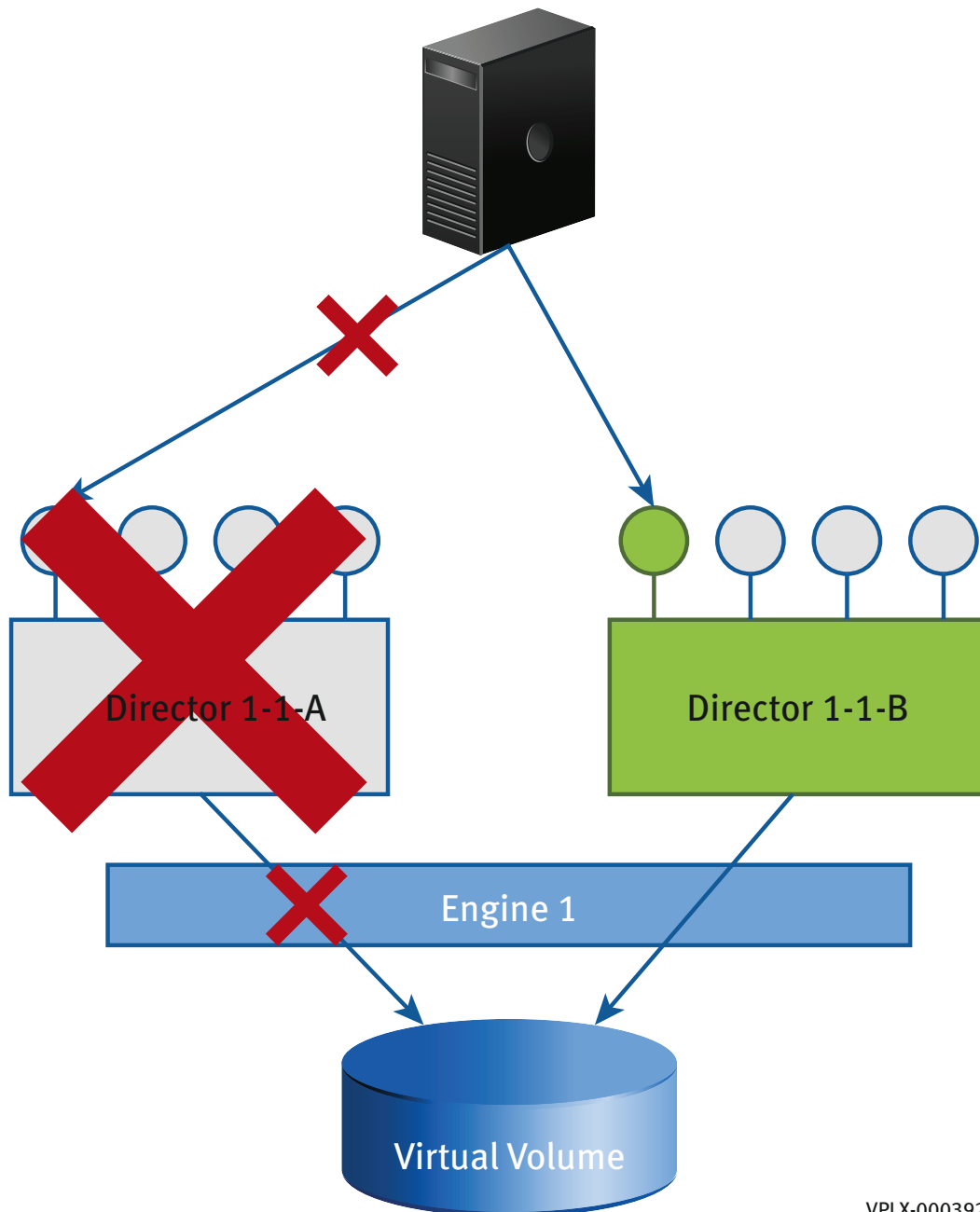
Combine multi-pathing software plus redundant volume presentation for continuous data availability in the presence of port failures.

Back-end ports, local COM ports, and WAN COM ports provide similar redundancy for additional resilience.

Each director can service I/O for any other director in the cluster due to the redundant nature of the global directory and cache coherency.

If one director in the engine fails, another director continues to service I/O from the host.

In the following figure, Director 1-1-A has failed, but Director 1-1-B services the host I/O that was previously being serviced by Director 1-1-A.



VPLX-000392

**Figure 13. Path redundancy: different directors**

## Management server

Each metro node server has embedded management server. You can manage both clusters in a metro node Metro configuration from a single management server. The management server acts as a management interfaces to other metro node components in the cluster. Redundant internal network IP interfaces connect the management server to the public network. Internally, the management server is on a dedicated management IP network that provides accessibility to all major components in the cluster.

The larger role of the management server includes:

- Coordinating data collection, metro node software upgrades, configuration interfaces, diagnostics, event notifications, and some director-to-director communication.

## Metro node Metro Hardware

To ensure continuous availability across multiple data centers in a metro region, metro node Metro provides an ideal solution with the option of Metro over IP (MetroIP).

Metro node use a metro node Metro with a 10 Gb Ethernet.

## Software and upgrade

This chapter describes the GeoSynchrony software that runs on the metro node hardware.

### Topics:

- [Metro node OS](#)
- [Non-disruptive upgrade \(NDU\)](#)

## Metro node OS

Metro node OS is the operating system that runs on the metro node hardware.

Metro node OS is:

- Designed for highly available, robust operation in geographically distributed environments
- Driven by real-time I/O operations
- Intelligent about locality of access
- Designed to provide the global directory that supports AccessAnywhere

The following table summarizes features provided by metro node OS and AccessAnywhere:

**Table 4. Metro node OS AccessAnywhere features**

Feature	Description and considerations
Storage volume encapsulation	LUNs on a back-end array can be imported into an instance of metro node and used while keeping their data intact.
	<b>Considerations:</b> The storage volume retains the existing data on the device and leverages the media protection and device characteristics of the back-end LUN.
RAID 1	metro node devices can be mirrored within a site.
	<b>Considerations:</b> Withstands a device failure within the mirrored pair. <ul style="list-style-type: none"> <li>• A device rebuild is a simple copy from the remaining device to the newly repaired device. Rebuilds are done in incremental fashion, whenever possible.</li> <li>• The number of required devices is twice the amount required to store data (actual storage capacity of a mirrored array is 50%).</li> <li>• The RAID 1 devices can come from different back-end array LUNs providing the ability to tolerate the failure of a back-end array.</li> </ul>
Distributed RAID 1	Metro node devices can be mirrored between sites.
	<b>Considerations:</b> Provides protection from site disasters and supports the ability to move data between geographically separate locations.
Migration	Volumes can be migrated non-disruptively to other storage systems.
	<b>Considerations:</b> Use migration for changing the quality of service of a volume or for performing technology refresh operations.

**Table 4. Metro node OS AccessAnywhere features (continued)**

Feature	Description and considerations
Global Visibility	The presentation of a volume from one metro node cluster where the physical storage for the volume is provided by a remote metro node cluster.
	<b>Considerations:</b> Use Global Visibility for AccessAnywhere collaboration between locations. The cluster without local storage for the volume will use its local cache to service I/O but non-cached operations incur remote latencies to write or read the data.

## Non-disruptive upgrade (NDU)

Metro node management server software and metro node OS can be upgraded without disruption.

Metro node hardware can be replaced, the engine count in a cluster increased, and a metro node Local can be expanded to metro node Metro without disruption.

Metro node never has to be completely shut down.

## Storage, application, and host upgrades

Metro node enables the easy addition or removal of storage, applications, and hosts.

When metro node encapsulates back-end storage, the block-level nature of the coherent cache allows the upgrade of storage, applications, and hosts.


You can configure metro node so that all devices within metro node have uniform access to all storage blocks.

## Software upgrades

Metro node is fully redundant for:

- Ports
- Paths
- Nodes

This redundancy allows GeoSynchrony on metro node Local and Metro to be upgraded without interrupting host access to storage, it does not require service window or application disruption.

 **NOTE:** You must upgrade the metro node management server software before upgrading GeoSynchrony. Management server upgrades are non-disruptive.

## Simple support matrix

Dell EMC publishes storage array interoperability information in a Simple Support Matrix available on Dell EMC Online Support. This information details tested, compatible combinations of storage hardware and applications that metro node supports. The Simple Support Matrix can be located at:

<https://www.dell.com/support>

## A

### **AccessAnywhere**

The breakthrough technology that enables metro node clusters to provide access to information between clusters that are separated by distance.

### **active/active**

A cluster with no primary or standby servers, because all servers can run applications and interchangeably act as backup for one another.

### **active/passive**

A powered component that is ready to operate upon the failure of a primary component.

### **Active Directory**

A directory service included in most Windows Server operating systems. AD authenticates and authorizes users and computers in a network of Windows domain type.

### **active mirror**

A copy of data that is part of a local or remote mirroring service.

## B

### **backend port**

Metro node director port connected to storage arrays (acts as an initiator).

### **bandwidth**

The range of transmission frequencies a network can accommodate, expressed as the difference between the highest and lowest frequencies of a transmission cycle. High bandwidth allows fast or high volume transmissions.

### **bias**

When a cluster has the bias for a given DR1, it continues to service I/O to volumes on that cluster if connectivity to the remote cluster is lost (due to cluster partition or cluster failure). The bias for a specific volume is determined by the detach rules for the volume, the detach rules for the consistency group (if the volume is a member of a consistency group) and metro node Witness (if metro node Witness is deployed).

### **bit**

A unit of information that has a binary digit value of either 0 or 1.

### **block**

The smallest amount of data that can be transferred following SCSI standards, which is traditionally 512 bytes. Virtual volumes are presented to users as a contiguous lists of blocks.

### **block size**

The actual size of a block on a device.

### **Bookmark**

A label applied to a snapshot so that the snapshot can be explicitly called (identified) during recovery processes (during image access). Bookmarks are created through the CLI or GUI and can be created manually, by the user, or automatically, by the system. Bookmarks created automatically can be created at

pre-defined intervals or in response to specific system events. Parallel bookmarks are bookmarks that are created simultaneously across multiple consistency groups.

### **byte**

Memory space used to store eight bits of data.



## C

### **cache coherency**

Managing the cache so that data is not lost, corrupted, or overwritten. With multiple processors, data blocks may have several copies, one in the main memory and one in each of the cache memories. Cache coherency propagates the blocks of multiple users throughout the system in a timely fashion, ensuring that the data blocks do not have inconsistent versions in the different processors caches.

### **cluster**

Two or more metro node directors forming a single fault-tolerant cluster, deployed as one to four engines.

### **cluster deployment ID**

A numerical cluster identifier, unique within a metro node cluster. By default, metro node clusters have a cluster deployment ID of 1. For multi-cluster deployments, all but one cluster must be reconfigured to have different cluster deployment IDs.

### **cluster ID**

The identifier for each cluster in a multi-cluster deployment. The ID is assigned during installation.

### **clustering**

Using two or more computers to function together as a single entity. Benefits include fault tolerance and load balancing, which increases reliability and up time.

### **cluster IP seed**

The metro node IP seed is used to generate the IP addresses used by the internal components of the metro node. For more information about components and their IP addresses, see the Dell EMC Installation and Setup Guide for metro node. Cluster ID is used by the virtualization software (inter director messaging, cluster identification).

## COM

The intra-cluster communication. The communication used for cache coherency and replication traffic.

### **command line interface (CLI)**

An interface that supports the use of typed commands to execute specific tasks.

### **consistency group**

A metro node structure that groups together virtual volumes and applies the same detach and failover rules to all member volumes. Consistency groups ensures the common application of a set of properties to the entire group. Create consistency groups for sets of volumes that require the same I/O behavior in the event of a link failure. There are two types of consistency groups:

- Synchronous Consistency Groups - Use write-through (synchronous) cache mode to write data to the underlying storage before an acknowledgment is sent to the host. This is dependent on the latency between clusters and the application's tolerance of the latency.
- Asynchronous Consistency Groups - Use write-back (asynchronous) cache mode to write protect data by mirroring it to the memory of another director in the cluster. Data is destaged asynchronously to the back-end storage arrays. Writes are acknowledged once the data has been committed to disk in write order.

### **continuity of operations (COOP)**

The goal of establishing policies and procedures to be used during an emergency, including the ability to process, store, and transmit data before and after.

### **controller**

A device that controls the transfer of data to and from a computer and a peripheral device.

## D

### **data sharing**

The ability to share access to the same data with multiple servers regardless of time and location.

**detach rule**

Predefined rules that determine which cluster continues I/O when connectivity between clusters is lost. A cluster loses connectivity to its peer cluster due to cluster partition or cluster failure.

Detach rules are applied at two levels; to individual volumes, and to consistency groups. If a volume is a member of a consistency group, the group detach rule overrides the rule set for the individual volumes. Note that all detach rules may be overridden by metro node Witness, if metro node Witness is deployed.

**device**

A combination of one or more extents to which you add specific RAID properties. Local devices use storage from only one cluster. In metro node Metro and Geo configurations, distributed devices use storage from both clusters.

**director**

A CPU module that runs metro node OS, the core metro node software. There are two directors (A and B) in each engine, and each has dedicated resources and is capable of functioning independently.

**dirty data**

The write-specific data stored in the cache memory that is yet to be written to disk.

**disaster recovery (DR)**

The ability to restart system operations after an error, preventing data loss.

**discovered array**

An array that is connected to the SAN and discovered by metro node.

**disk cache**

A section of RAM that provides cache between the disk and the CPU. RAMs access time is significantly faster than disk access time. Therefore, a disk-caching program enables the computer to operate faster by placing recently accessed data in the disk cache.

**Distributed consistency groups**

The RecoverPoint consistency group is divided into four segments. Each segment runs on one primary RPA and one to three secondary RPAs.

Distributed consistency groups enable a much higher throughput and IOPS rate, regardless of the amount of data being replicated.

**distributed device**

A RAID 1 device whose mirrors are in different metro node clusters.

**distributed file system (DFS)**

Supports the sharing of files and resources in the form of persistent storage over a network.

**distributed RAID1 device (DR1)**

Distributed devices have physical volumes at both clusters in a metro node Metro configuration for simultaneous active/active and read/write access using AccessAnywhere.

**E****Ethernet**

A Local Area Network (LAN) protocol. Ethernet uses a bus topology, meaning all devices are connected to a central cable, and supports data transfer rates of between 10 megabits per second and 10 gigabits per second. For example, 100 Base-T supports data transfer rates of 100 Mb/s.

**event**

A log message that results from a significant action initiated by a user or the system.

**extent**

All or a portion (range of blocks) of a storage volume.

## F

### **failover**

Automatically switching to a redundant or standby device, system, or data path upon the failure or abnormal termination of the currently active device, system, or data path.

### **fault domain**

A set of components that share a single point of failure. For metro node, the concept that every component of a highly available system is separated, so that if a fault occurs in one domain, it will not result in failure in other domains to which it is connected.

### **fault tolerance**

Ability of a system to keep working in the event of hardware or software failure, usually achieved by duplicating key system components.

### **Fibre Channel (FC)**

A protocol for transmitting data between computer devices. Longer distance requires the use of optical fiber. However, FC also works using coaxial cable and ordinary telephone twisted pair media. Fibre channel offers point-to-point, switched, and loop interfaces. Used within a SAN to carry SCSI traffic.

### **Fibre Channel over IP (FCIP)**

Combines Fibre Channel and Internet protocol features to connect SANs in geographically distributed systems.

### **field replaceable unit (FRU)**

A unit or component of a system that can be replaced on site as opposed to returning the system to the manufacturer for repair.

### **firmware**

Software that is loaded on and runs from the flash ROM on the metro node directors.

### **front end port**

Metro node director port connected to host initiators (acts as a target).

## G

### **geographically distributed system**

A system that is physically distributed across two or more geographically separated sites. The degree of distribution can vary widely, from different locations on a campus or in a city to different continents.

### **gigabit (Gb or Gbit)**

1,073,741,824 ( $2^{30}$ ) bits. Often rounded to  $10^9$ .

### **gigabit Ethernet**

The version of Ethernet that supports data transfer rates of 1 gigabit per second.

### **gigabyte (GB)**

1,073,741,824 ( $2^{30}$ ) bytes. Often rounded to  $10^9$ .

### **global file system (GFS)**

A shared-storage cluster or distributed file system.

## H

### **hold provisioning**

An attribute of a registered array that allows you to set the array as unavailable for further provisioning of new storage.

### **host bus adapter (HBA)**

An I/O adapter that manages the transfer of information between the host computers bus and memory system. The adapter performs many low-level interface functions automatically or with minimal processor involvement to minimize the impact on the host processors performance.

## **I**

### **infiniband**

A networking standard used for passing data between and among computers. The VS6 hardware uses this protocol for the intra-cluster communication.

### **input/output (I/O)**

Any operation, program, or device that transfers data to or from a computer.

### **internet Fibre Channel protocol (iFCP)**

Connects Fibre Channel storage devices to SANs or the Internet in geographically distributed systems using TCP.

### **internet small computer system interface (iSCSI)**

A protocol that allows commands to travel through IP networks, which carries data from storage units to servers anywhere in a computer network.

### **intranet**

A network operating like the World Wide Web but with access restricted to a limited group of authorized users.

## **J**

### **Journal volumes**

Volumes that contain data waiting to be distributed to target replica volumes and copies of the data previously distributed to the target volumes. Journal volumes allow convenient rollback to any point in time, enabling instantaneous recovery for application environments.

## **K**

### **kilobit (Kb)**

1,024 ( $2^{10}$ ) bits. Often rounded to  $10^3$ .

### **kilobyte (KB)**

1,024 ( $2^{10}$ ) bytes. Often rounded to  $10^3$ .

## **L**

### **latency**

Amount of time required to fulfill an I/O request.

### **LDAP**

Lightweight Directory Access Protocol, an application protocol that accesses and maintains distributed directory information services over an IP network.

### **load balancing**

Distributing the processing and communications activity evenly across a system or network so that no single device is overwhelmed. Load balancing is especially important when the number of I/O requests issued is unpredictable.

### **local area network (LAN)**

A group of computers and associated devices that share a common communications line and typically share the resources of a single processor or server within a small geographic area.

**local device**

A combination of one or more extents to which you add specific RAID properties. Local devices use storage from only one cluster.

**logical unit number (LUN)**

Virtual storage to which a given server with a physical connection to the underlying storage device may be granted or denied access. LUNs are used to identify SCSI devices, such as external hard drives that are connected to a computer. Each device is assigned a LUN number which serves as the device's unique address.

## M

**Management Module Control Station (MMCS)**

The management entity (management server) is VS6 hardware. The first engine in a cluster has two MMCS: A and B. All the remaining engines have Akula management modules for the management connectivity.

**megabit (Mb)**

1,048,576 ( $2^{20}$ ) bits. Often rounded to  $10^6$ .

**megabyte (MB)**

1,048,576 ( $2^{20}$ ) bytes. Often rounded to  $10^6$ .

**metavolume**

A storage volume used by the system that contains the metadata for all the virtual volumes managed by the system. There is one metadata storage volume per cluster.

**Metro-Plex**

Two metro node Metro clusters connected within Metro (synchronous) distances, approximately 60 miles or 100 kilometers.

**MetroPoint consistency group (Metro group)**

A consistency group that is protecting a DR1 volume in a MetroPoint topology.

**mirroring**

The writing of data to two or more disks simultaneously. If one of the disk drives fails, the system can instantly switch to one of the other disks without losing data or service. RAID 1 provides mirroring.

**mirroring services**

Mirroring features provided through a storage service profile.

**miss**

An operation where the cache is searched but does not contain the data, so the data instead must be accessed from disk.

## N

**namespace**

A set of names recognized by a file system in which all names are unique.

**network**

System of computers, terminals, and databases connected by communication lines.

**network architecture**

Design of a network, including hardware, software, method of connection, and the protocol used.

**network-attached storage (NAS)**

Storage elements connected directly to a network.

**network partition**

When one site loses contact or communication with another site.

## **Non-distributed consistency groups**

Transfer data through one primary RPA that is designated by the user during group creation. The policies applied by the consistency group can be modified at any time.

In the event of RPA failure, groups that transfer data through the failed RPA will move to other RPAs in the cluster.

## **O**

### **Open LDAP**

Open source implementation of the Lightweight Directory Access Protocol (LDAP).

## **P**

### **parity checking**

Checking for errors in binary data. Depending on whether the byte has an even or odd number of bits, an extra 0 or 1 bit, called a parity bit, is added to each byte in a transmission. The sender and receiver agree on odd parity, even parity, or no parity. If they agree on even parity, a parity bit is added that makes each byte even. If they agree on odd parity, a parity bit is added that makes each byte odd. If the data is transmitted incorrectly, the change in parity will reveal the error.

### **partition**

A subdivision of a physical or virtual disk, which is a logical entity only visible to the end user and not to any of the devices.

### **point-in-time (PIT)**

See the description for Snapshot/PIT.

### **Production journal volumes**

Volumes that hold system delta marking information.

### **Production volumes**

Volumes that are written to by the host applications. Writes to production volumes are split such that they are sent to both the normally designated volumes and RPAs simultaneously.

Each production volume must be exactly the same size as the replica volume to which it replicates.

## **R**

### **RAID**

The use of two or more storage volumes to provide better performance, error recovery, and fault tolerance.

#### **RAID 0**

A performance orientated striped or dispersed data mapping technique. Uniformly sized blocks of storage are assigned in regular sequence to all of the arrays disks. Provides high I/O performance at low inherent cost. No additional disks are required. The advantages of RAID 0 are a very simple design and an ease of implementation.

#### **RAID 1**

Also called mirroring, this has been used longer than any other form of RAID. It remains popular because of simplicity and a high level of data availability. A mirrored array consists of two or more disks. Each disk in a mirrored array holds an identical image of the user data. RAID 1 has no striping. Read performance is improved since either disk can be read at the same time. Write performance is lower than single disk storage. Writes must be performed on all disks, or mirrors, in the RAID 1. RAID 1 provides very good data reliability for read-intensive applications.

#### **RAID leg**

A copy of data, called a mirror, that is located at a user's current location.

### **rebuild**

The process of reconstructing data onto a spare or replacement drive after a drive failure. Data is reconstructed from the data on the surviving disks, assuming mirroring has been employed.

**RecoverPoint Appliance (RPA)**

Hardware that manages all aspects of data protection for a storage group, including capturing changes, maintaining the images in the journal volumes, and performing image recovery.

**RecoverPoint cluster**

All connected RecoverPoint Appliances on both sides of the replication.

**RecoverPoint site**

All RecoverPoint entities that are on one side of the replication.

**Recovery Point Objective (RPO)**

Recovery Point Objective. The time interval between the point of failure of a storage system and the expected point in the past at which the storage system is capable of recovering customer data. Informally, RPO is maximum amount of data loss that can be tolerated by the application after a failure. The value of the RPO is highly dependent upon the recovery technique used. For example, RPO for backups is typically days, for asynchronous replication minutes, and for mirroring or synchronous replication seconds or instantaneous.

**Recovery Time Objective (RTO)**

Recovery Time Objective. Not to be confused with RPO, RTO is the time duration within which a storage solution is expected to recover from failure and begin servicing application requests. Informally, RTO is the longest tolerable application outage due to a failure of a storage system. RTO is a function of the storage technology. It may measure in hours for backup systems, minutes for a remote replication, and seconds (or less) for a mirroring.

**redundancy**

The duplication of hardware and software components. In a redundant system, if a component fails then a redundant component takes over, allowing operations to continue without interruption.

**registered array**

An array that is registered with metro node. Registration is required to make the array available for services-based provisioning. Registration includes connecting to and creating awareness of the array's intelligent features. Only VMAX and VNX arrays can be registered.

**reliability**

The ability of a system to recover lost data.

**Remote Direct Memory Access (RDMA)**

Allows computers within a network to exchange data using their main memories and without using the processor, cache, or operating system of either computer.

**Replica journal volumes**

Volumes that hold:

- Snapshots that are either waiting to be replicated or already distributed to the replica
- Metadata for each image
- Bookmarks

Each replica journal holds as many snapshots as its capacity allows.

After distribution, the oldest snapshot is discarded to make room for the newest snapshot. The number of snapshots in the journal depends on the size of the snapshots and the capacity of the volumes.

**replication set**

When RecoverPoint is deployed, a production source volume and one or more replica volumes to which it replicates.

**replica volumes**

Volumes to which production volumes replicate. In prior releases, the replica volume must be exactly the same size as its production volume. In RecoverPoint (RP) 4.0 and GeoSynchrony release 5.2, RecoverPoint supports a feature called Fake Size, where the replica volume size can be higher than the production volume.

**repository volume**

A volume dedicated to RecoverPoint for each RPA cluster. The repository volume serves all RPAs of the particular RPA cluster and the splitter associated with that cluster. The repository volume stores configuration information about the RPAs and RecoverPoint consistency groups. There is one repository volume per RPA cluster.

**restore source**

This operation restores the source consistency group from data on the copy target.

## S

**scalability**

Ability to easily change a system in size or configuration to suit changing conditions, to grow with your needs.

**simple network management protocol (SNMP)**

Monitors systems and devices in a network.

**site ID**

The identifier for each cluster in a multi-cluster metro node. By default, in a non-geographically distributed system the ID is 0. In a geographically distributed system, the ID of the first cluster is 1, the next is 2, and so on, each number identifying a physically separate cluster. These identifiers are assigned during installation.

**SLES**

SUSE Linux Enterprise Server (SLES) is a Linux distribution supplied by SUSE and targeted at the business market.

**small computer system interface (SCSI)**

A set of evolving ANSI standard electronic interfaces that allow personal computers to communicate faster and more flexibly than previous interfaces with peripheral hardware such as disk drives, tape drives, CD-ROM drives, printers, and scanners.

**Snapshot/PIT**

A point-in-time copy that preserves the state of data at an instant in time, by storing only those blocks that are different from an already existing full copy of the data.

Snapshots are also referred to as point-in-time (PIT). Snapshots stored at a replica journal represent the data that has changed on the production storage since the closing of the previous snapshot.

**splitter**

Dell EMC RecoverPoint write-splitting technology built into GeoSynchrony starting in 5.1.

**storage area network (SAN)**

A high-speed special purpose network or subnetwork that interconnects different kinds of data storage devices with associated data servers on behalf of a larger network of users.

**storage view**

A combination of registered initiators (hosts), front-end ports, and virtual volumes, used to control a hosts access to storage.

**storage volume**

A Logical Unit Number (LUN) or unit of storage presented by the back end array.

**stripe depth**

The number of blocks of data stored contiguously on each storage volume in a RAID 0 device.

**striping**

A technique for spreading data over multiple disk drives. Disk striping can speed up operations that retrieve data from disk storage. Data is divided into units and distributed across the available disks. RAID 0 provides disk striping.

**synchronous**

Describes objects or events that are coordinated in time. A process is initiated and must be completed before another task is allowed to begin.



For example, in banking, two withdrawals from a checking account that started at the same time must not overlap; therefore, they are processed synchronously.

## T

### **throughput**

1. The number of bits, characters, or blocks passing through a data communication system or portion of that system.
2. The maximum capacity of a communications channel or system.
3. A measure of the amount of work performed by a system over a period of time. For example, the number of I/Os per day.

### **tool command language (TCL)**

A scripting language often used for rapid prototypes and scripted applications.

### **transfer size**

The size of the region in cache used to service data migration. The area is globally locked, read at the source, and written at the target. Transfer size can be as small as 40 K, as large as 128 M, and must be a multiple of 4 K. The default value is 128 K.

A larger transfer size results in higher performance for the migration, but may negatively impact front-end I/O. This is especially true for metro node Metro migrations. Set a large transfer-size for migrations when the priority is data protection or migration performance.

A smaller transfer size results in lower performance for the migration, but creates less impact on front-end I/O and response times for hosts. Set a smaller transfer-size for migrations when the priority is front-end storage response time.

### **transmission control protocol/Internet protocol (TCP/IP)**

The basic communication language or protocol used for traffic on a private network and the Internet.

## U

### **uninterruptible power supply (UPS)**

A power supply that includes a battery to maintain power in the event of a power failure.

### **universal unique identifier (UUID)**

A 64-bit number used to uniquely identify each metro node director. This number is based on the hardware serial number assigned to each director.

## V

### **virtualization**

A layer of abstraction that is implemented in software that servers use to divide available physical storage into storage volumes or virtual volumes.

### **virtual volume**

Unit of storage presented by the metro node front end ports to hosts. A virtual volume looks like a contiguous volume, but can be distributed over two or more storage volumes.

## W

### **wide area network (WAN)**

A geographically dispersed telecommunications network. This term distinguishes a broader telecommunication structure from a local area network (LAN).

### **world wide name (WWN)**

A specific Fibre Channel Name Identifier that is unique worldwide and represented by a 64-bit unsigned binary value.

**write-through mode**

A caching technique in which the completion of a write request is communicated only after data is written to disk. This is almost equivalent to non-cached systems, but with data protection.

# Index

## A

AccessAnywhere [17](#)  
ALUA [19](#)  
API [13](#)  
architecture [11](#)  
audience [7](#)  
availability [23](#)

## B

back-end load [21](#)  
backup metadata volumes [25](#)  
Big Data [17](#)

## C

CAW [22](#)  
certificates [19](#)  
CLI [13](#)  
cluster [11](#), [24](#)  
clusters [12](#), [23](#)  
collaboration [17](#)  
command line management [13](#)  
comments [7](#)  
configuration management [25](#)  
conventions for publication [7](#)  
CPU Load [21](#)

## D

director [26](#)

## E

Element Manager API [13](#)

## F

failures [23](#)  
faults [24](#)  
front-end load [21](#)

## H

HTTPS [19](#)

## I

integrity [23](#)  
IPsec [19](#)

## L

LDAP [19](#)  
load balancing [21](#)  
logging volumes [25](#)

## M

management from clusters [12](#)  
Management GUI [12](#)  
management server [28](#)  
metadata volumes [25](#)  
metro node Metro  
    Metro over IP [29](#)  
metro node OS [30](#)  
migration [16](#)  
mirrors [25](#)  
mobility [14](#), [16](#)  
monitoring [21](#), [22](#)  
monitoring CLI [22](#)  
monitors [22](#)  
multi-pathing [23](#)

## N

non-disruptive upgrade [31](#)

## O

outages [23](#)

## P

passwords [19](#)  
path optimization [19](#)  
performance [21](#), [22](#)  
port usage [19](#)  
preface [7](#)  
provisioning storage [20](#)

## Q

quorum [24](#)

## R

redundancy [23](#), [24](#)  
related documentation [7](#)  
resilience [25](#)  
resiliance [23](#)  
REST [13](#)

## S

security [19](#)  
site distribution [23](#)  
statistics [22](#)  
support information [7](#)

## T

technology refresh [16](#)  
thin volume [20](#)

## U

Unisphere for VPLEX [12](#)  
Unisphere GUI [12](#)  
Unisphere monitoring tool [21](#)  
unmap [20](#)  
upgrade [31](#)  
user roles [19](#)

## V

VPLEX hardware [26](#)  
VPLEX hardware platforms [11](#)  
VPLEX Witness [23](#)

## W

WAN link load [21](#)  
WWN [11](#)