

Array-based Replication with EMC VPLEX

Best Practices Planning



Abstract

This white paper provides best practices planning and use cases for using array-based replication solutions with EMC® VPLEX™ Local and EMC VPLEX Metro.

August 2010

Copyright © 2010 EMC Corporation. All rights reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com

All other trademarks used herein are the property of their respective owners.

Part Number h8005

Table of Contents

Executive summary	4
Document scope and limitations	4
Introduction	4
Audience	4
Terminology	5
VPLEX technology	5
EMC VPLEX architecture	6
EMC VPLEX family	7
EMC VPLEX clustering architecture	7
Array-based replication with VPLEX	8
Use Case 1: Basic array-based clones with VPLEX	9
Use Case 2: Advanced array-based clones with VPLEX	11
Prerequisites	12
Use Case 3: Basic array-based restore and recovery with VPLEX.....	14
Use Case 4: Advanced array-based restore with VPLEX	16
Prerequisites	17
Conclusion	19
References	19

Executive summary

The EMC® VPLEX™ family removes physical barriers within, across, and between data centers. VPLEX Local provides simplified management and nondisruptive data mobility across heterogeneous arrays. VPLEX Metro provides data access and mobility between two VPLEX clusters within synchronous distances. With a unique scale-up architecture, VPLEX's advanced data caching and distributed cache coherency provides workload resiliency, automatic sharing, and balancing and failover of storage domains, and enables both local and remote data access with predictable service levels.

With EMC VPLEX's new capabilities, users must take several important considerations into account when using array-based replication technologies. This paper explores array-based replication use cases and illustrates procedural steps you should take to properly account for VPLEX. As VPLEX Metro and Local leverage a write-through cache technology, additional planning is required. Today, this cache is exclusively used to accelerate read performance for VPLEX. Any time there is array write activity occurring outside of the VPLEX I/O path, the VPLEX read cache must be considered. Proper planning for VPLEX read cache allows for the preservation of existing array-based replication functionality.

Document scope and limitations

VPLEX is constantly evolving as a platform. The procedures and technology discussed in this white paper are only applicable to the VPLEX Local and VPLEX Metro products running Geosynchrony™ 4.1 and earlier. In particular, the procedures described only apply to the write-through caching versions of VPLEX. This paper will be updated as new software features and functionality are added to the VPLEX platform. Please consult with your local EMC support representative if you are uncertain as to the applicability of these procedures to your VPLEX environment.

Introduction

This white paper includes the following sections:

- VPLEX technology
- Array-based replication with VPLEX

The focus of this white paper is the best practices and key considerations for leveraging array-based replication technologies with VPLEX Local or Metro. We review the technical impact of VPLEX on current replication practices and provide use case examples. Each example illustrates the necessary procedural adjustments to ensure the expected results when using array-based replication. Use cases include:

- Use Case 1: Basic array-based clones with VPLEX
- Use Case 2: Advanced array-based clones with VPLEX
- Use Case 3: Basic array-based restore and recovery with VPLEX
- Use Case 4: Advanced array-based restore with VPLEX

Audience

This white paper is intended for technology architects, storage administrators, and system administrators who are responsible for architecting, creating, managing, and using the IT environments that utilize EMC VPLEX technologies. The white paper assumes that the reader is familiar with EMC VPLEX and storage array-based replication technologies.

Terminology

Table 1. Operational definitions

Term	Definition
Storage volume	LUN or unit of storage presented by the back-end arrays
Metadata volume	System volume that contains metadata about the devices, virtual volumes, and cluster configuration
Extent	All or part of a storage volume.
Device	Protection scheme applied to an extent or group of extents.
Virtual volume	Unit of storage presented by the VPLEX front-end ports to hosts
Front-end port	Director port connected to host initiators (acts as a target)
Back-end port	Director port connected to storage arrays (acts as an initiator)
Director	The central processing and intelligence of the VPLEX solution. There are redundant (A and B) directors in each VPLEX Engine
Engine	Consists of two directors and is the unit of scale for the VPLEX solution
VPLEX cluster	A collection of VPLEX engines in one rack, using redundant, private Fibre Channel connections as the cluster interconnect
Metro-Plex	A cooperative set of two VPLEX Metro clusters, each serving their own storage domain

Table 2. Acronyms and abbreviations

Acronym/Abbreviation	Definition
Clone	An independent array-based full disk copy of another device
Source device	Standard or primary array device. Typically used to run production applications
FE port	Front-end (target ports visible to hosts)
BE port	Back-end (initiator ports visible to storage arrays)

VPLEX technology

EMC VPLEX encapsulates traditional physical storage array devices and applies three layers of logical abstraction to them. The logical relationships of each layer are shown in Figure 1.

Extents are the mechanism VPLEX uses to divide storage volumes. Extents may be all or part of the underlying storage volume. EMC VPLEX aggregates extents and applies RAID protection in the device layer. Devices are constructed using one or more extents and can be combined into more complex RAID schemes and device structures as desired. At the top layer of the VPLEX storage structures are virtual volumes. Virtual volumes are created from devices and inherit the size of underlying device. Virtual volumes are the elements VPLEX exposes to hosts using its FE ports. Access to virtual volumes is controlled using storage views. Storage views are comparable to Auto-provisioning Groups on EMC Symmetrix® or to storage groups on EMC CLARiiON®. They act as logical containers determining host initiator access to VPLEX FE ports and virtual volumes.

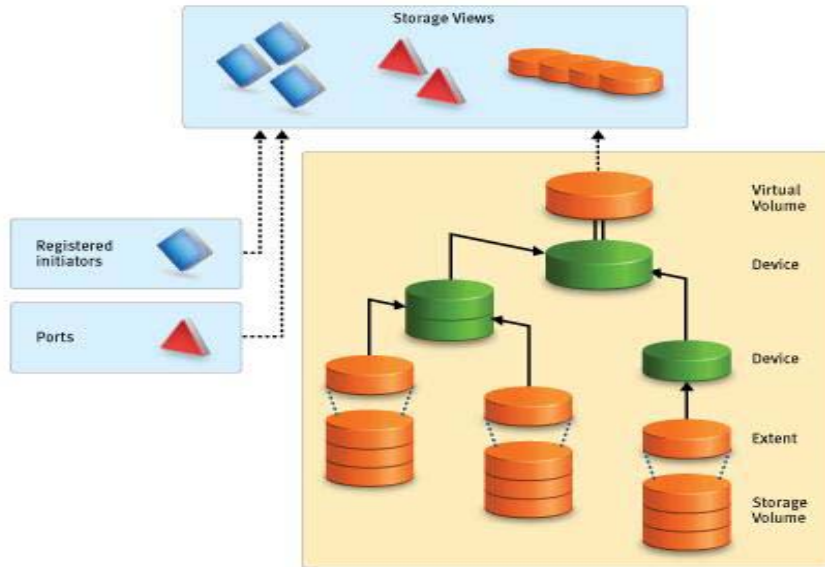


Figure 1. EMC VPLEX logical storage structures

EMC VPLEX architecture

EMC VPLEX represents the next-generation architecture for data mobility and information access. The new architecture is based on EMC's more than 20 years of expertise in designing, implementing, and perfecting enterprise-class intelligent cache and distributed data protection solutions.

As shown in Figure 2, VPLEX is a solution for federating both EMC and non-EMC storage. VPLEX resides between servers and heterogeneous storage assets and introduces a new architecture with these unique characteristics:

- Scale-up clustering hardware, which lets customers start small and grow big with predictable service levels
- Advanced data caching, which utilizes large-scale SDRAM cache to improve performance and reduce I/O latency and array contention
- Distributed cache coherence for automatic sharing, balancing, and failover of I/O across the cluster
- A consistent view of one or more LUNs across VPLEX clusters separated either by a few feet within a data center or across synchronous distances, enabling new models of high availability and workload relocation

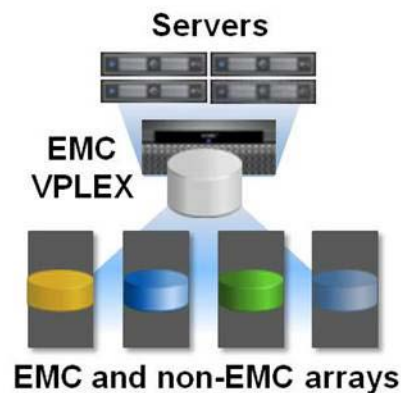


Figure 2. Capability of an EMC VPLEX system to federate heterogeneous storage

EMC VPLEX family

The EMC VPLEX family consists of two offerings:

- **VPLEX Local:** This solution is appropriate for customers who would like to federate homogeneous or heterogeneous storage systems within a data center and manage data mobility between the physical data storage entities.
- **VPLEX Metro:** This solution is for customers who require concurrent access and data mobility across two locations separated by synchronous distances. The VPLEX Metro offering also includes the unique capability where a remote VPLEX Metro site can present LUNs without the need for physical storage for those LUNs at the remote site.

The EMC VPLEX family with current architectural limits is shown in Figure 3.

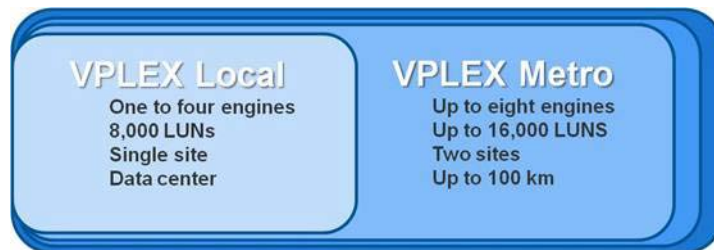


Figure 3. EMC VPLEX family offering with architectural limits

EMC VPLEX clustering architecture

VPLEX uses a unique clustering architecture to help customers break the boundaries of the data center and allow servers at multiple data centers to have concurrent read and write access to shared block storage devices. A VPLEX cluster, shown in Figure 4, can scale up through the addition of more engines, and scale out by connecting multiple clusters to form a VPLEX Metro configuration. In the initial release, a VPLEX Metro system supports up to two clusters, which can be in the same data center, or at two different sites within synchronous distances (approximately up to 60 miles or 100 kilometers apart). VPLEX Metro configurations help users to transparently move and share workloads, consolidate data centers, and optimize resource utilization across data centers. In addition, VPLEX clusters provide nondisruptive data mobility, heterogeneous storage management, and improved application availability.

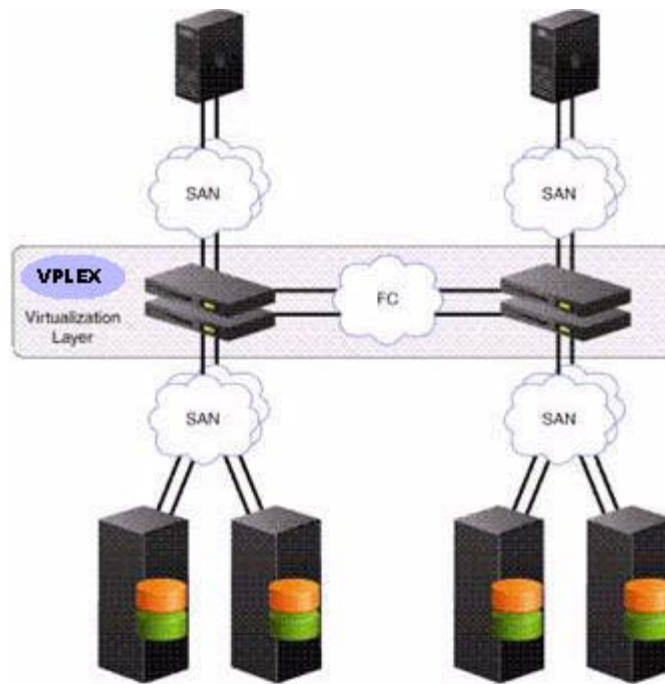


Figure 4. Schematic representation of supported EMC VPLEX cluster configurations

A VPLEX cluster is composed of one, two, or four engines. The engine is responsible for virtualizing the I/O stream, and connecting to hosts and storage using Fibre Channel connections as the data transport. A single VPLEX cluster consists of one to four engines.

Each engine is composed of:

- Two directors, which run the Geosynchrony software and connect to storage, hosts, and other directors in the cluster with Fibre Channel and gigabit Ethernet connections
- One Standby Power Supply, which provides backup power to sustain the engine through transient power loss
- Two management modules, which contain interfaces for remote management of a VPLEX Engine

Each cluster also consists of:

- A management server, which manages the cluster and provides an interface from a remote management station
- An EMC standard 40U cabinet to hold all of the cluster's equipment

Additionally, clusters containing more than one engine also have:

- A pair of Fibre Channel switches used for inter-director communication between various engines
- A pair of Universal Power Supplies that provide backup power for the Fibre Channel switches and allow the system to ride through transient power loss.

Array-based replication with VPLEX

Preserving the investments made in array-based storage replication is of critical importance in today's IT environment. EMC designs with this goal in mind and the VPLEX product family is no exception. By following the VPLEX best practices outlined here, array-based replication tools can continue delivering expected functionality and value within your IT infrastructure. As you will see, one of the ways VPLEX preserves array replication technologies is by mapping storage volumes in their entirety (one-to-one mapping) through VPLEX. When done in this fashion, the underlying storage devices are left untouched by VPLEX and the back-end array LUN replication technology continues to function normally.

An important key to each of the following VPLEX use case examples is the read cache invalidation step. Per-volume read cache invalidation happens when VPLEX virtual volumes are removed from storage views. The act of removing a VPLEX virtual volume from a storage view causes the read cache for the corresponding volume to be invalidated (discarded). This is of the utmost importance with array-based replication and restore due to the fact that the array is writing to the back-end storage device outside of the VPLEX I/O path. When this happens the VPLEX read cache does not match the data on the physical disk on the array. This has the potential to cause data corruption. To avoid this scenario, it is critical that each VPLEX virtual volume is removed from the storage view it is a member of while array-based (non-VPLEX I/O path writes) are taking place. Once the replication activities involving writes are completed and the underlying back-end storage is in the desired state, then the virtual volume can be added back into a storage view and be accessed normally.

Use Case 1: Basic array-based clones with VPLEX

The simplest use case for array-based replication with VPLEX is a clone or full disk array-based copy. Clones can be presented through VPLEX to a backup host, test environment, or even back to the original host. The setup for this use case is illustrated in Figure 5. This figure examines both the initial presentation of the array-based clones through VPLEX and the subsequent re-synchronization of the clones. We assume each array-based clone has a one-to-one storage volume pass-through configuration (device capacity = extent capacity = storage volume capacity) to VPLEX and has a RAID 0 (single extent only) VPLEX device geometry.

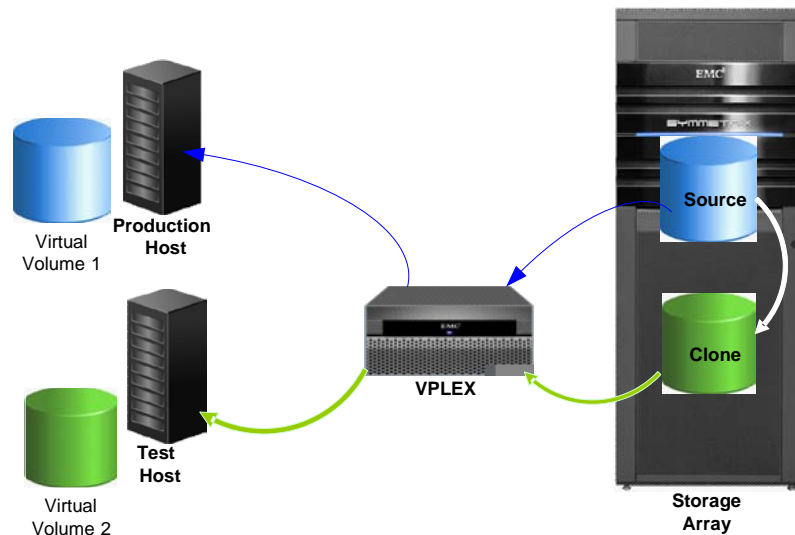


Figure 5. Basic array-based clones with VPLEX

Follow these steps to use an array-based clone of a VPLEX virtual volume.

To perform standard array-based copy procedure(s):

1. Within the array, identify the source storage volumes from which you want to make array-based clones from.
2. Follow array-based cloning procedure(s) to generate the desired array-based clones. Reference your specific array-based replication documentation for exact commands and procedures. For EMC Symmetrix, see the *Solutions Enabler Symmetrix TimeFinder Family CLI 7.x Product Guide* and the *Solutions Enabler Symmetrix SRDF Family CLI 7.x Product Guide*. For EMC CLARiiON see, the *SnapView Command Line Interface (CLI) Reference Rev 17* and the *MirrorView Command Line Interface (CLI) Reference A17*.

To perform VPLEX-specific steps:

3. Confirm that the clone devices are visible to VPLEX. As necessary, perform array LUN masking and SAN zoning for storage volumes containing clones to VPLEX back-end ports.

4. Perform one-to-one encapsulation through VPLEX:
 - a. Claim storage volumes
 - b. Create extents (using the `-appc` flag)
 - c. Create devices (single extent RAID 0 geometry)
 - d. Create virtual volumes for each array-based clone
5. Present virtual volumes based on clones to host(s).
 - a. If necessary, create storage view(s).
 - b. Add virtual volumes built from array-based copies to storage view(s).
 - c. If necessary, perform zoning of virtual volumes to hosts following normal zoning procedures.

To resynchronize array-based clones once they've been presented through VPLEX, follow these steps:

1. Shut down any applications using the VPLEX array-based clones and, if necessary, unmount the associated virtual volumes to be resynchronized.
2. Remove access to the virtual volumes constructed from array-based clones. Be sure to note the LUN number for the virtual volume you plan to remove.

```
/clusters/<cluster name>/exports/storage-views> removevirtualvolume -v storage_view_name -o virtual_volume_name -f
```

3. Wait 30 seconds to ensure that the VPLEX read cache has been invalidated for each virtual volume. This can be done concurrently with step 4. Perform normal array-based resynchronization procedure(s).
4. Identify the source storage volumes within the array you wish to resynchronize. Follow your normal array resynchronization procedure(s) to refresh the desired array-based clones.
5. Confirm the IO Status of storage volumes based on array-based clones is “alive” by doing a long listing against the storage-volumes context for your cluster.

For example:

```
VPlexcli:/> ll /clusters/cluster-1/storage-elements/storage-volumes/
```

/clusters/cluster-1/storage-elements/storage-volumes:									
Name	VPD83 ID	Capacity	Use	Vendor	IO Status	Type			
Symm1554Tdev_061D	VPD83T3:60000970000192601554533030363144	100G	used	EMC	alive	normal			
Symm1554Tdev_061E	VPD83T3:60000970000192601554533030363145	100G	claimed	EMC	alive	normal			
Symm1554Tdev_061F	VPD83T3:60000970000192601554533030363146	100G	claimed	EMC	alive	normal			
Symm1554Tdev_0620	VPD83T3:60000970000192601554533030363230	100G	claimed	EMC	alive	normal			
Symm1554Tdev_0621	VPD83T3:60000970000192601554533030363231	100G	claimed	EMC	alive	normal			
Symm1554Tdev_0622	VPD83T3:60000970000192601554533030363232	100G	used	EMC	alive	normal			
Symm1554Tdev_0623	VPD83T3:60000970000192601554533030363233	100G	used	EMC	alive	normal			
Symm1554Tdev_0624	VPD83T3:60000970000192601554533030363234	100G	used	EMC	alive	normal			
Symm1554Tdev_0625	VPD83T3:60000970000192601554533030363235	100G	used	EMC	alive	normal			
Symm1554Tdev_0626	VPD83T3:60000970000192601554533030363236	100G	claimed	EMC	alive	normal			

In addition, confirm VPLEX back-end paths are healthy by issuing the “connectivity validate-be” command from the VPLEX CLI. Ensure that there are no errors or connectivity issues to the back-end storage devices. Resolve any error conditions with the back-end storage before proceeding.

Example output showing desired back-end status:

```

Vplexcli:/clusters/cluster-1> connectivity validate-be
Summary
Cluster cluster-2
  This cluster has 0 storage-volumes which are dead or unreachable
  This cluster has 0 storage-volumes which do not have dual paths
  This cluster has 0 storage-volumes which are not visible from all directors
Cluster cluster-1
  This cluster has 0 storage-volumes which are dead or unreachable
  This cluster has 0 storage-volumes which do not have dual paths
  This cluster has 0 storage-volumes which are not visible from all directors

```

6. Restore access to virtual volumes based on clone devices for host(s). Add the virtual volume back to the view, specifying the original LUN number (noted in step 2) using the VPLEX CLI:

```

/clusters/<cluster name>/exports/storage-views> addvirtualvolume -v
storage_view_name/ -o (lun#, virtual_volume_name) -f

```

7. Rescan devices and restore paths (for example, powermt restore) on hosts.
8. If necessary, mount devices.
9. Restart applications.

Note: VPLEX does not change the rules for array-based clones. When presenting an array-based clone back to the same host, you will need to confirm both the host operating system and the logical volume manager to support such an operation. If the host OS does not support cloning and representation of the same volume, VPLEX will not change this situation.

Use Case 2: Advanced array-based clones with VPLEX

When VPLEX virtual volumes have RAID 1 geometry, the array-based cloning process becomes more complex. For this use case, we assume a RAID 1 device layout for VPLEX consisting of one mirror leg that is an array-based clone and a second mirror leg that is not a clone. Figure 6 illustrates the advanced array-based clone configuration use case. This configuration applies to both local (VPLEX Local) RAID 1 and distributed (VPLEX Metro) RAID 1 devices. Using array-based clones in this type of setup requires few tweaks to the standard clone creation and clone resynchronization processes. These additional steps are critical to ensure proper mirror synchronization (for the non-clone leg) and to ensure each virtual volume's read cache is properly updated. Figure 6 illustrates the case of an array-based clone that is one leg of a distributed RAID 1 (two-leg) volume.

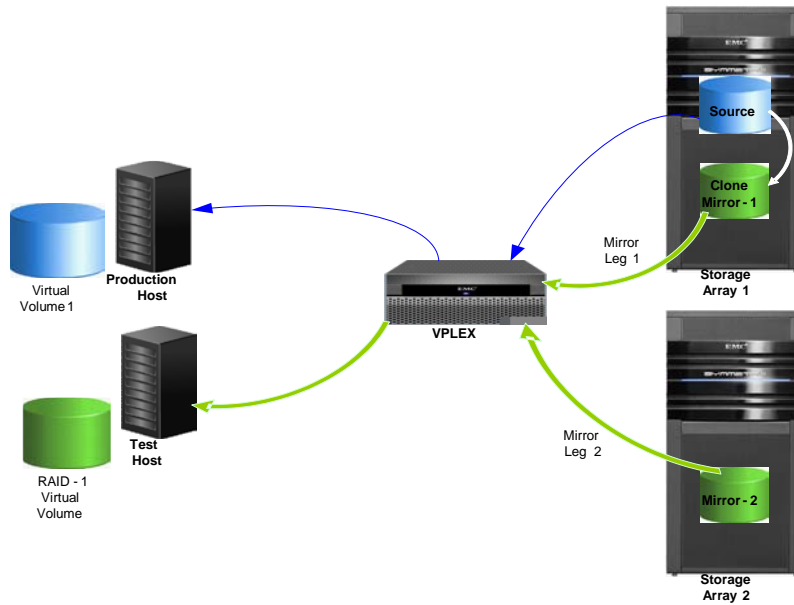


Figure 6. Advanced array-based clones with RAID 1 VPLEX storage volumes

Prerequisites

This section assumes you are using existing distributed or local RAID 1 VPLEX virtual volumes built from storage volumes that are array-based copies. In addition, the VPLEX virtual volumes must possess both of the following attributes:

- Be comprised of devices that have a one-to-one storage volume pass-through configuration to VPLEX (device capacity = extent capacity = storage volume capacity).
- Have a single device with single-extent RAID 1 (two single-extent devices being mirrored) geometry.

Follow these steps to use an array-based copy of a RAID 1 VPLEX virtual volume:

1. Quiesce the application environment following normal local array disk restore preparation procedures. This step is of particular importance as all paths to the VPLEX volumes being restored will become temporarily unavailable during subsequent steps. The goal is to have no read or write activity to array-based copies where the VPLEX volumes reside during the initial synchronization and subsequent resynchronization processes.
2. Remove host access to corresponding VPLEX volumes by removing them from all storage views.
 - a) If the virtual volume is built from a local device and/or is a member of a single storage view, using the VPLEX CLI run:

```
/clusters/<cluster name>/exports/storage-views> removevirtualvolume -v
storage_view_name -o virtual_volume_name -f
```

- b) If the virtual volume is built from a distributed device and is a member of storage views in both clusters, using the VPLEX CLI run:

```
/clusters/<local cluster name>/exports/storage-views> removevirtualvolume -v
storage_view_name -o distributed_device_name_vol -f
```

```
/clusters/<remote cluster name>/exports/storage-views> removevirtualvolume -v
storage_view_name -o distributed_device_name_vol -f
```

Note: It is vital to keep track of the VPLEX-assigned LUN number for each virtual volume you plan to resynchronize. You can obtain this information by executing a long listing (ll) in the corresponding storage view context from the VPLEX CLI, or by clicking on the storage view from the VPLEX Management Console.

3. Detach the VPLEX device mirror leg that will *not* be updated during the array-based replication or resynchronization processes:

- a) For distributed RAID 1 devices turn off logging:

```
/distributed-storage/distributed-devices> set-log -n -d /distributed-storage/distributed-devices/distributed_device_name_vol
```

- b) Detach the mirror leg:

```
device detach-mirror -m <device_mirror_to_detach> -d <distributed_device_name> -i -f
```

4. Perform replication or resynchronization of array-based copies.
5. Confirm the IO Status of storage volumes based on array-based clones is “alive” by doing a long listing against the storage-volumes context for your cluster.

For example:

```
VPlexcli:/> ll /clusters/cluster-1/storage-elements/storage-volumes/
```

/clusters/cluster-1/storage-elements/storage-volumes:							
Name	VPD63 ID	Capacity	Use	Vendor	IO Status	Type	
Symm1554Tdev_061D	VPD83T3:60000970000192601554533030363144	100G	used	EMC	alive	normal	
Symm1554Tdev_061E	VPD83T3:60000970000192601554533030363145	100G	claimed	EMC	alive	normal	
Symm1554Tdev_061F	VPD83T3:60000970000192601554533030363146	100G	claimed	EMC	alive	normal	
Symm1554Tdev_0620	VPD83T3:60000970000192601554533030363230	100G	claimed	EMC	alive	normal	
Symm1554Tdev_0621	VPD83T3:60000970000192601554533030363231	100G	claimed	EMC	alive	normal	
Symm1554Tdev_0622	VPD83T3:60000970000192601554533030363232	100G	used	EMC	alive	normal	
Symm1554Tdev_0623	VPD83T3:60000970000192601554533030363233	100G	used	EMC	alive	normal	
Symm1554Tdev_0624	VPD83T3:60000970000192601554533030363234	100G	used	EMC	alive	normal	
Symm1554Tdev_0625	VPD83T3:60000970000192601554533030363235	100G	used	EMC	alive	normal	
Symm1554Tdev_0626	VPD83T3:60000970000192601554533030363236	100G	claimed	EMC	alive	normal	

In addition, confirm VPLEX back-end paths are healthy by issuing the “connectivity validate-be” command from the VPLEX CLI. Ensure that there are no errors or connectivity issues to the back-end storage devices. Resolve any error conditions with the back-end storage before proceeding.

Example output showing desired back-end status:

```
VPlexcli:/clusters/cluster-1> connectivity validate-be
```

```
Summary
```

```
Cluster cluster-2
```

```
  This cluster has 0 storage-volumes which are dead or unreachable
```

```
  This cluster has 0 storage-volumes which do not have dual paths
```

```
  This cluster has 0 storage-volumes which are not visible from all directors
```

```
Cluster cluster-1
```

```
  This cluster has 0 storage-volumes which are dead or unreachable
```

```
  This cluster has 0 storage-volumes which do not have dual paths
```

```
  This cluster has 0 storage-volumes which are not visible from all directors
```

6. Reattach the second mirror leg:
 - a) Attach the mirror:

```
device attach-mirror -m <2nd_mirror_leg_to_attach> -d
/clusters/local_cluster_name/devices/existing_raid_1_device
```

b) Turn logging back on for distributed devices:

```
/distributed-storage/distributed-devices> set-log -d /distributed-
storage/distributed-devices/distributed_raid_1_device_1
```

Note: The device you are attaching is the non-clone mirror leg. It will be overwritten with the data from the clone mirror leg.

7. Restore host access to VPLEX volume(s).

If the virtual volume is built from a local RAID 1 device:

```
/clusters/<local cluster name>/exports/storage-views> addvirtualvolume -v
storage_view_name/ -o (lun#,device_Symm0191_065_1_vol/) -f
```

If the virtual volume is built from a distributed RAID 1 device:

```
/clusters/<remote cluster name>/exports/storage-views> addvirtualvolume -v
storage_view_name/ -o (lun#, distributed_device_name_vol)-f
```

```
/clusters/<local cluster name>/exports/storage-views> addvirtualvolume -v
storage_view_name/ -o (lun#, distributed_device_name_vol)-f
```

The lun# is the previously recorded value from step 2 for each virtual volume.

Note: EMC recommends waiting at least 30 seconds after removing access from a storage view to restore access. Waiting ensures that the VPLEX cache has been cleared for the volumes. The array-based resynchronization will likely will take 30 seconds, but if you are scripting, be sure to add a pause prior to performing this step.

Some hosts and applications are sensitive to LUN numbering changes. Use the information you recorded in step 3 to ensure that you use the same LUN numbering when the virtual volume access is restored.

You do not need to perform a full mirror synchronization prior to restoring access to virtual volumes. VPLEX will synchronize the second mirror leg in the background while using the first mirror leg as necessary to service reads to any unsynchronized blocks.

8. Rescan devices and restore paths (powermt restore) on hosts.

9. Mount devices (if mounts are used).

10. Restart applications.

Use Case 3: Basic array-based restore and recovery with VPLEX

The array-based restore process with VPLEX is similar to the previous array-based clone use cases. The primary difference is that the array is now writing data from the clone or “gold copy” back to the primary production or source volume. This use case examines the array-based restore process and assumes the array-based clone or gold copy is accessible from the array containing the source device (restore target). It is further assumed that each source device has a one-to-one storage volume pass-through configuration (device capacity = extent capacity = storage volume capacity) to VPLEX *and a RAID 0 (single extent only) device geometry*. This is the most basic use case for array-based restore and recovery. Figure 7 illustrates the case where data is being written within a storage array from a clone (copy) or backup media to a storage volume (standard device) that is used by VPLEX.

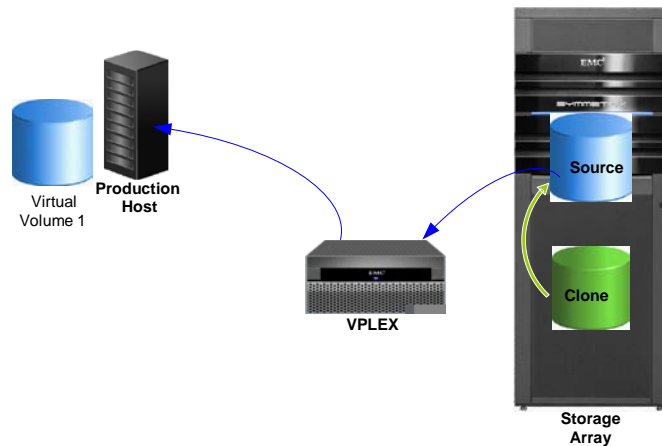


Figure 7. Basic array-based restore of VPLEX storage volumes

Follow these steps to perform an array-based restore to a VPLEX virtual volume:

1. Shut down any host applications using the VPLEX source devices and, as necessary, unmount the associated virtual volumes you need to restore.
2. Remove host access to the virtual volumes constructed from source devices. Be sure to note the LUN number for the virtual volume you plan to remove.


```
/clusters/<cluster name>/exports/storage-views> removevirtualvolume -v storage_view_name -o virtual_volume_name -f
```
3. Wait 30 seconds to ensure that the VPLEX read cache has been invalidated for each virtual volume. This can be done concurrently with step 4.
4. Perform normal array-based restore and/or recovery procedure(s). Identify the clones or gold copies to restore to the source devices with the array. Follow your normal array restore procedure(s) to refresh the desired source devices.
5. Confirm the IO Status of storage volumes based on array-based clones is “alive” by doing a long listing against the storage volumes context for your cluster.

For example:

```
Vplexcli:/> ll /clusters/cluster-1/storage-elements/storage-volumes/

/clusters/cluster-1/storage-elements/storage-volumes:
Name                               VPD83 ID                               Capacity Use      Vendor  IO Status  Type
-----
Symm1554Tdev_061D                  VPD83T3:60000970000192601554533030363144 100G   used     EMC     alive     normal
Symm1554Tdev_061E                  VPD83T3:60000970000192601554533030363145 100G   claimed  EMC     alive     normal
Symm1554Tdev_061F                  VPD83T3:60000970000192601554533030363146 100G   claimed  EMC     alive     normal
Symm1554Tdev_0620                  VPD83T3:60000970000192601554533030363230 100G   claimed  EMC     alive     normal
Symm1554Tdev_0621                  VPD83T3:60000970000192601554533030363231 100G   claimed  EMC     alive     normal
Symm1554Tdev_0622                  VPD83T3:60000970000192601554533030363232 100G   used     EMC     alive     normal
Symm1554Tdev_0623                  VPD83T3:60000970000192601554533030363233 100G   used     EMC     alive     normal
Symm1554Tdev_0624                  VPD83T3:60000970000192601554533030363234 100G   used     EMC     alive     normal
Symm1554Tdev_0625                  VPD83T3:60000970000192601554533030363235 100G   used     EMC     alive     normal
Symm1554Tdev_0626                  VPD83T3:60000970000192601554533030363236 100G   claimed  EMC     alive     normal
```

In addition, confirm VPLEX back-end paths are healthy by issuing the “connectivity validate-be” command from the VPLEX CLI. Ensure that there are no errors or connectivity issues to the back-end storage devices. Resolve any error conditions with the back-end storage before proceeding.

Example output showing desired back-end status:

```
VPlexcli:/clusters/cluster-1> connectivity validate-be
Summary
Cluster cluster-2
  This cluster has 0 storage-volumes which are dead or unreachable
  This cluster has 0 storage-volumes which do not have dual paths
  This cluster has 0 storage-volumes which are not visible from all directors
Cluster cluster-1
  This cluster has 0 storage-volumes which are dead or unreachable
  This cluster has 0 storage-volumes which do not have dual paths
  This cluster has 0 storage-volumes which are not visible from all directors
```

6. Restore access to virtual volume(s) based on source devices for host(s): Add the virtual volume back to the view, specifying the original LUN number (noted in step 2) using VPLEX CLI:

```
/clusters/<cluster name>/exports/storage-views> addvirtualvolume -v
storage_view_name/ -o (lun#, virtual_volume_name) -f
```

7. Rescan devices and restore paths (for example, powermt restore) on hosts.
8. If necessary, mount devices.
9. Restart applications.

Use Case 4: Advanced array-based restore with VPLEX

When VPLEX virtual volumes have RAID 1 geometry, the restore process must take into account this added complexity. This applies for both local (VPLEX Local) RAID 1 and distributed (VPLEX Metro) RAID 1 VPLEX devices. The typical array-based source device restore only restores one of the two mirror legs of a VPLEX RAID 1 device. In order to synchronize the second VPLEX device, users need to modify standard array-based restore procedures. These steps are critical to ensure proper synchronization of the second VPLEX device mirror leg (the one that is not part of the array-based restore) and to ensure each virtual volume's read cache is properly updated. Figure 8 illustrates the case when an array-based clone is used to restore to a distributed RAID 1 source volume. **This same use case applies to a remote array-based copy being restored to a source volume.**

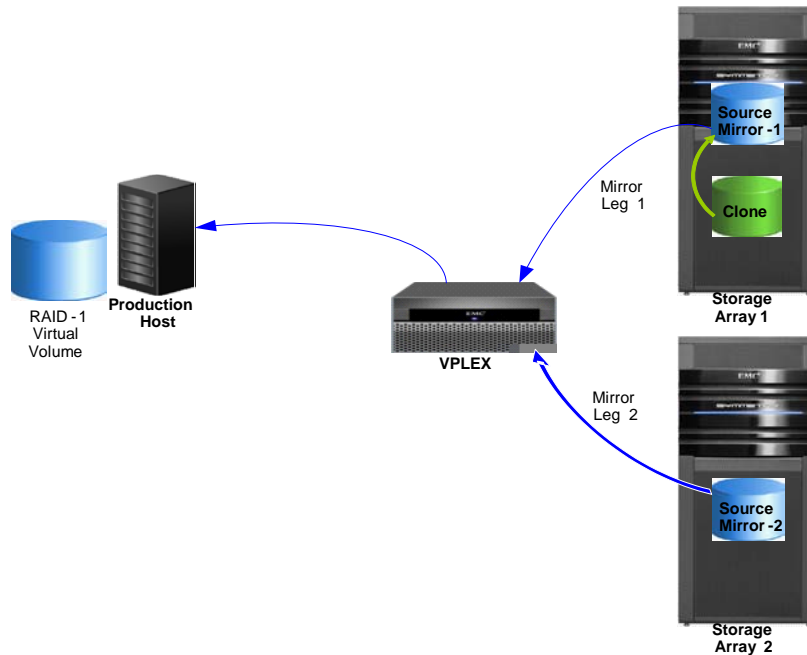


Figure 8. Advanced array-based restore and recovery of RAID 1 VPLEX volumes

Prerequisites

This section assumes users have existing distributed or local RAID 1 VPLEX virtual volumes built from the array source devices being restored. In addition, the VPLEX virtual volumes must possess both of the following attributes:

- The volumes must comprise of devices that have a one-to-one storage volume pass-through VPLEX configuration (device capacity = extent capacity = storage volume capacity).
- The volumes must have a single-extent RAID-1 (two single extents being mirrored) geometry.

Follow the next steps to perform an array-based restore to a RAID 1 VPLEX virtual volume:

1. Quiesce the application environment following normal local array disk restore preparation procedures. This step is of particular importance as all paths to the VPLEX virtual volumes being restored will become temporarily unavailable during subsequent steps. The goal is to have no read or write activity to array-based copies where the VPLEX volumes reside during the initial synchronization and subsequent resynchronization processes.
2. Remove host access to corresponding VPLEX volumes by removing them from *all* storage views.
 - a) If the virtual volume is built from a local device and/or is a member of a single storage view, using the VPLEX CLI run:

```
/clusters/<cluster name>/exports/storage-views> removevirtualvolume -v
storage_view_name -o virtual_volume_name -f
```

- b) If the virtual volume is built from a distributed device and is a member of storage views in both clusters, using the VPLEX CLI run:

```
/clusters/<local cluster name>/exports/storage-views> removevirtualvolume -v
storage_view_name -o distributed_device_name_vol -f
```

```
/clusters/<remote cluster name>/exports/storage-views> removevirtualvolume -v
storage_view_name -o distributed_device_name_vol -f
```

Note: It is vital to keep track of the VPLEX-assigned LUN number for each virtual volume you plan to restore. You can obtain this information by executing a long listing (ll) in the corresponding storage view context from the VPLEX CLI or by clicking on the host storage view from the VPLEX Management Console.

3. Detach the VPLEX device mirror leg that will *not* be restored during the array-based restore processes:

a) For distributed RAID 1 devices turn off logging:

```
/distributed-storage/distributed-devices> set-log -n -d /distributed-storage/distributed-devices/distributed_device_name_vol
```

b) Detach the mirror leg:

```
device detach-mirror -m <device_mirror_to_detach> -d <distributed_device_name> -i -f
```

4. Perform the array-based clone device to source device restore process.

5. Confirm the IO Status of storage volumes based on array-based clones is “alive” by doing a long listing against the storage volumes context for your cluster.

For example:

```
Vplexcli:/> ll /clusters/cluster-1/storage-elements/storage-volumes/

/clusters/cluster-1/storage-elements/storage-volumes:
Name                               VPD83 ID                               Capacity Use      Vendor  IO Status  Type
-----
Symm1554Tdev_061D                  VPD83T3: 60000970000192601554533030363144 100G   used     EMC     alive     normal
Symm1554Tdev_061E                  VPD83T3: 60000970000192601554533030363145 100G   claimed  EMC     alive     normal
Symm1554Tdev_061F                  VPD83T3: 60000970000192601554533030363146 100G   claimed  EMC     alive     normal
Symm1554Tdev_0620                  VPD83T3: 60000970000192601554533030363230 100G   claimed  EMC     alive     normal
Symm1554Tdev_0621                  VPD83T3: 60000970000192601554533030363231 100G   claimed  EMC     alive     normal
Symm1554Tdev_0622                  VPD83T3: 60000970000192601554533030363232 100G   used     EMC     alive     normal
Symm1554Tdev_0623                  VPD83T3: 60000970000192601554533030363233 100G   used     EMC     alive     normal
Symm1554Tdev_0624                  VPD83T3: 60000970000192601554533030363234 100G   used     EMC     alive     normal
Symm1554Tdev_0625                  VPD83T3: 60000970000192601554533030363235 100G   used     EMC     alive     normal
Symm1554Tdev_0626                  VPD83T3: 60000970000192601554533030363236 100G   claimed  EMC     alive     normal
```

In addition, confirm VPLEX back-end paths are healthy by issuing the “connectivity validate-be” command from the VPLEX CLI. Ensure that there are no errors or connectivity issues to the back-end storage devices. Resolve any error conditions with the back-end storage before proceeding.

Example output showing desired back-end status:

```
Vplexcli:/clusters/cluster-1> connectivity validate-be
Summary
Cluster cluster-2
  This cluster has 0 storage-volumes which are dead or unreachable
  This cluster has 0 storage-volumes which do not have dual paths
  This cluster has 0 storage-volumes which are not visible from all directors
Cluster cluster-1
  This cluster has 0 storage-volumes which are dead or unreachable
  This cluster has 0 storage-volumes which do not have dual paths
  This cluster has 0 storage-volumes which are not visible from all directors
```

6. Reattach the second mirror leg:

a) Attach the mirror:

```
device attach-mirror -m <2nd_mirror_leg_to_attach> -d
  /clusters/local_cluster_name/devices/existing_raid_1_device
```

b) Turn logging back on for distributed devices:

```
/distributed-storage/distributed-devices> set-log -d /distributed-  
storage/distributed-devices/distributed_raid_1_device_1
```

Note: The device you are attaching in this step will be overwritten with the data from the newly restored source device.

7. Restore host access to the VPLEX volume(s).

If the virtual volume is built from a local RAID 1 device:

```
/clusters/<local cluster name>/exports/storage-views> addvirtualvolume -v  
storage_view_name/ -o (lun#,device_Symm0191_065_1_vol/) -f
```

If the virtual volume is built from a distributed RAID 1 device:

```
/clusters/<remote cluster name>/exports/storage-views> addvirtualvolume -v  
storage_view_name/ -o (lun#, distributed_device_name_vol)-f
```

```
/clusters/<local cluster name>/exports/storage-views> addvirtualvolume -v  
storage_view_name/ -o (lun#, distributed_device_name_vol)-f
```

The **lun#** is the previously recorded value from step 2 for each virtual volume.

Note: EMC recommends waiting at least 30 seconds after removing access from a storage view to restore access. This is done to ensure that the VPLEX cache has been cleared for the volumes. The array-based restore will likely take 30 seconds, but if you are scripting be sure to add a pause.

Some hosts and applications are sensitive to LUN numbering changes. Use the information you recorded in step 3 to ensure the same LUN numbering when you restore the virtual volume access.

Full mirror synchronization is not required prior to restoring access to virtual volumes. VPLEX will synchronize the second mirror leg in the background while using the first mirror leg as necessary to service reads to any unsynchronized blocks.

8. Rescan devices and restore paths (powermt restore) on hosts.

9. Mount devices (if mounts are used).

10. Restart applications.

Conclusion

Array-based replication can continue to provide the same value it does today in an EMC VPLEX environment when appropriate modifications to existing procedures are made.

References

- [EMC VPLEX page](#) on EMC.com
- [Workload Resiliency with EMC VPLEX](#) white paper
- [EMC VPLEX Architecture and Deployment: Enabling the Journey to the Private Cloud](#) TechBook