PURESTORAGE®

PURE STORAGE
PURITY ACTIVECLUSTER

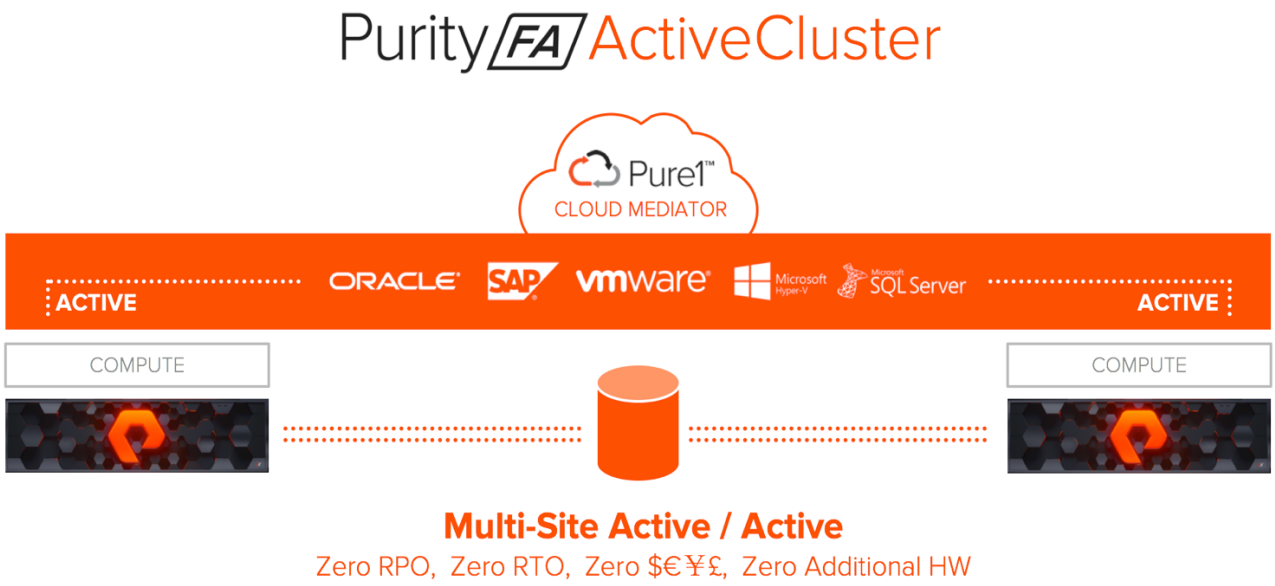
Solution Overview

Contents

Overview

Introduction

Pure Storage® Purity ActiveCluster is a fully symmetric active/active bidirectional replication solution that provides synchronous replication for RPO zero and automatic transparent failover for RTO zero. ActiveCluster spans multiple sites enabling clustered arrays and clustered hosts to be used to deploy flexible active/active datacenter configurations.



Synchronous Replication - Writes are synchronized between arrays and protected in non-volatile RAM (NVRAM) on both of the arrays before being acknowledged to the host.

Symmetric Active-Active - Read and write to the same volumes at either side of the mirror, with optional host-to-array site awareness.

Transparent Failover with Preferences – Automatic Non-disruptive failover between synchronously replicating arrays and sites according to user selected preference with automatic resynchronization and recovery.

Active-Active Asynchronous Replication - Integrated Target-Orchestrated asynchronous replication provides resilient and dramatically simplified out of region (3rd site) data protection.

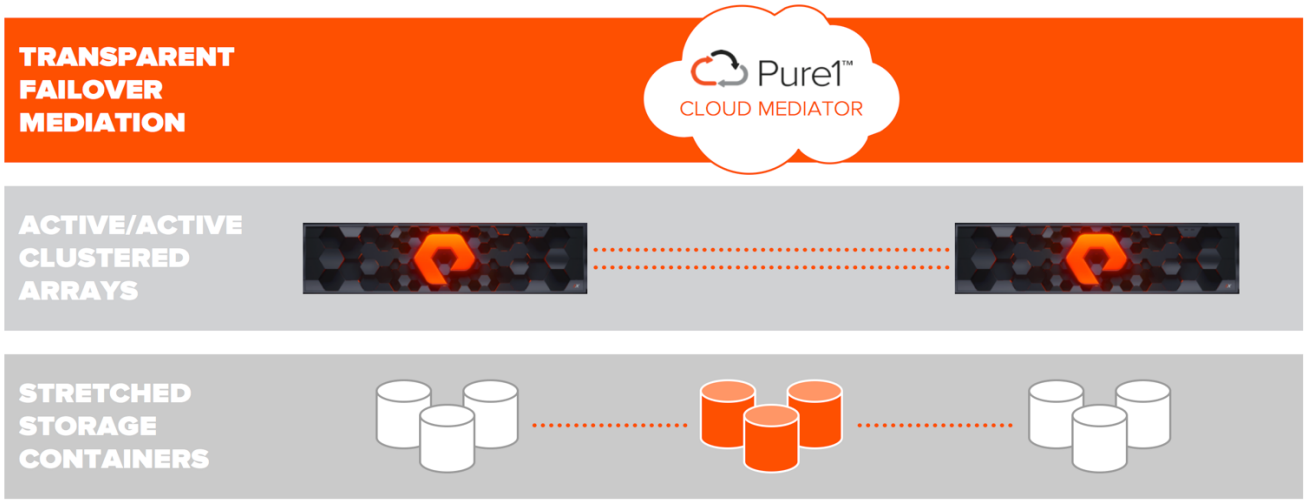
No Bolt-Ons & No Licenses - No additional hardware required, no costly software licenses required, just upgrade the Purity Operating Environment and go active/active!

Simple Management - Perform data management operations from either side of the mirror, provision storage, connect hosts, create snapshots, create clones.

Integrated Pure1® Cloud Mediator - Automatically configured passive mediator that allows transparent failover and prevents split-brain, without the need to deploy and manage another component.

Core Components

Purity ActiveCluster is composed of three core components: The Pure1 Mediator, active/active clustered array pairs, and stretched storage containers.



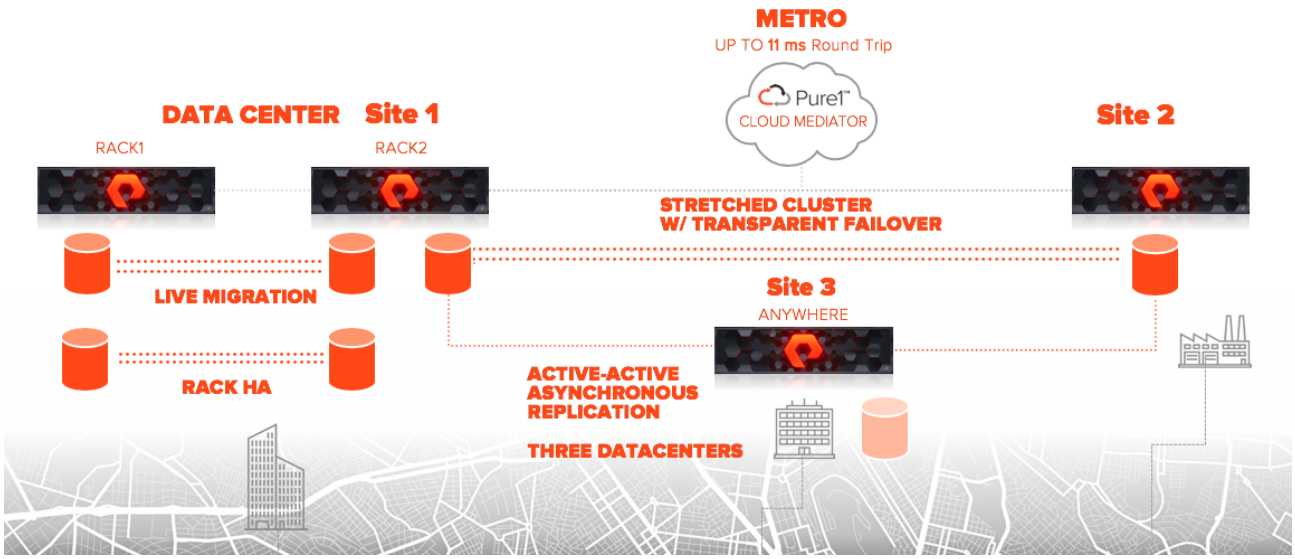
The Pure1 Cloud Mediator - A required component of the solution that is used to determine which array will continue data services should an outage occur in the environment.

Active-Active Clustered FlashArrays - Utilize synchronous replication to maintain a copy of data on each array and present those as one consistent copy to hosts that are attached to either, or both, arrays.

Stretched Storage Containers - Management containers that collect storage objects such as volumes into groups that are stretched between two arrays. Stretched storage containers also provide consistent IO continuation behavior for the storage objects within them.

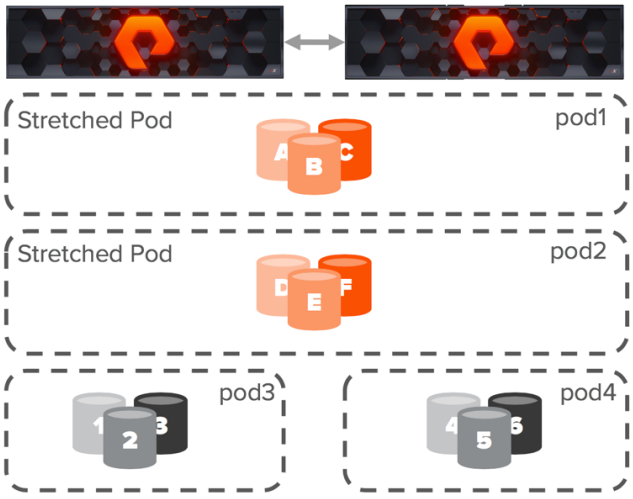
Use Cases

ActiveCluster is an extremely flexible FlashArray feature which enables use cases within and across datacenters. Within a single datacenter, ActiveCluster provides storage high availability across multiple FlashArrays, non-disruptive application and data mobility, as well as, workload consolidation from one array to another. Across multiple data centers, ActiveCluster provides synchronous replication and enables stretched clustering with transparent failover (0 RPO/RTO), data center migration, as well as seamless integration with out-of-region asynchronous replication.



Administration

ActiveCluster introduces a new management object: Pods. A Pod is a stretched storage container that defines a set of objects that are synchronously replicated together, and which arrays they are replicated between. An array can support multiple Pods. Pods can exist on just one array or on two arrays simultaneously with synchronous replication. Pods that are synchronously replicated between two arrays are said to be stretched between arrays.



Pods can contain volumes, protection groups (for snapshot scheduling and asynchronous replication) and other configuration information such as which volumes are connected to which hosts. The pod acts as a consistency group, ensuring that multiple volumes within the same pod remain write order consistent.

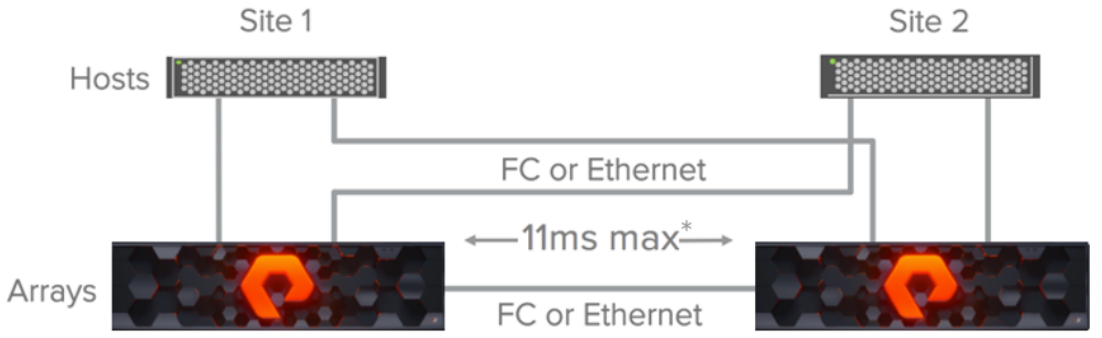
Pods also provide volume namespaces, that is different volumes may have the same volume name if they are in different pods. In the image above the volumes in pod3 and pod 4 are different volumes than those in pod1, a stretched active/active pod. This allows migration of workloads between arrays or consolidation of workloads from multiple arrays to one, without volume name conflicts.

Deployment Options

ActiveCluster storage (volumes) can be accessed by hosts using either a uniform or a non-uniform SAN connectivity topology. In a uniform storage access configuration, each host has connectivity to both ActiveCluster FlashArrays. In a non-uniform storage configuration, each host connects to just one ActiveCluster FlashArray. There is no need to configure ActiveCluster for one way or the for other; it is determined simply by the way hosts are connected to the ActiveCluster.

Uniform Storage Access

A uniform storage access model can be used in environments where there is host-to-array connectivity of either FC or ethernet (for iSCSI), and array-to-array ethernet connectivity, between the two sites. When deployed in this way a host has access to the same volume through both the local array and the remote array. The solution supports connecting arrays with up to 11ms of round-trip time (RTT) latency between the arrays.



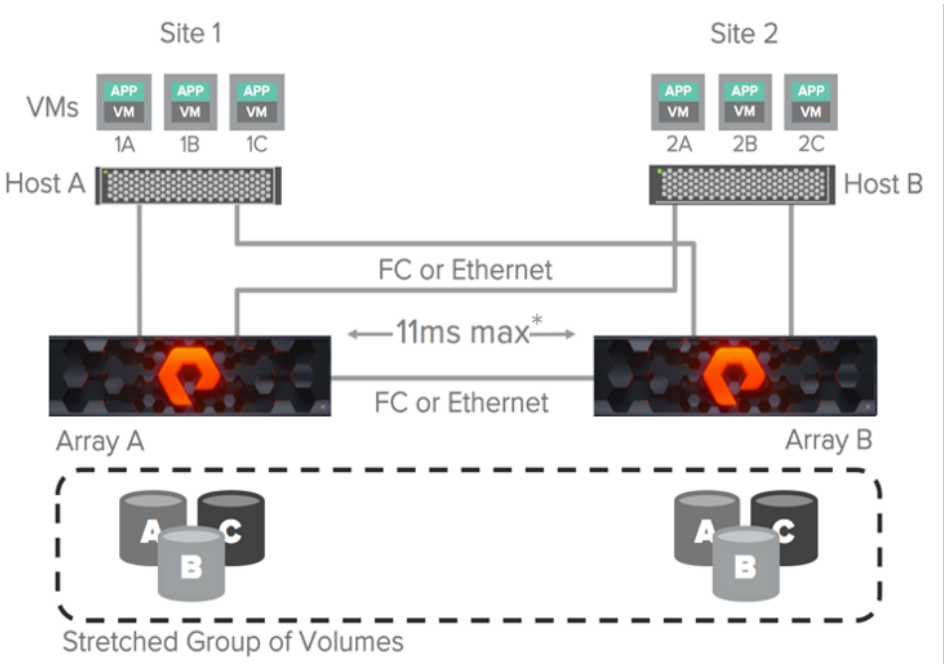
*SUBJECT TO PROTOCOL LIMITS. SEE THE Purity Replication Requirements KB FOR THE MOST CURRENT INFORMATION.

The image above represents the logical paths that exist between the hosts and arrays, and the replication connection between the two arrays in a uniform access model. Because a uniform storage access model allows all hosts, regardless of site location, to access both arrays there will be paths with different latency characteristics. Paths from hosts to the local array will have lower latency; paths from each local host to the remote array will have higher latency.

Optimizing Performance in Uniform Access Environments

For the best performance in an active/active synchronous replication environments hosts should be prevented from using paths that access the remote array unless necessary. For example, in the image below if VM 2A were to perform a write to volume A over the host side connection to array A, that write would incur 2X the latency of the inter site link, 1X for each traverse of the network. The write would

experience 11ms of latency for the trip from host B to array A and experience another 11ms of latency while array A synchronously sends the write back to array B.

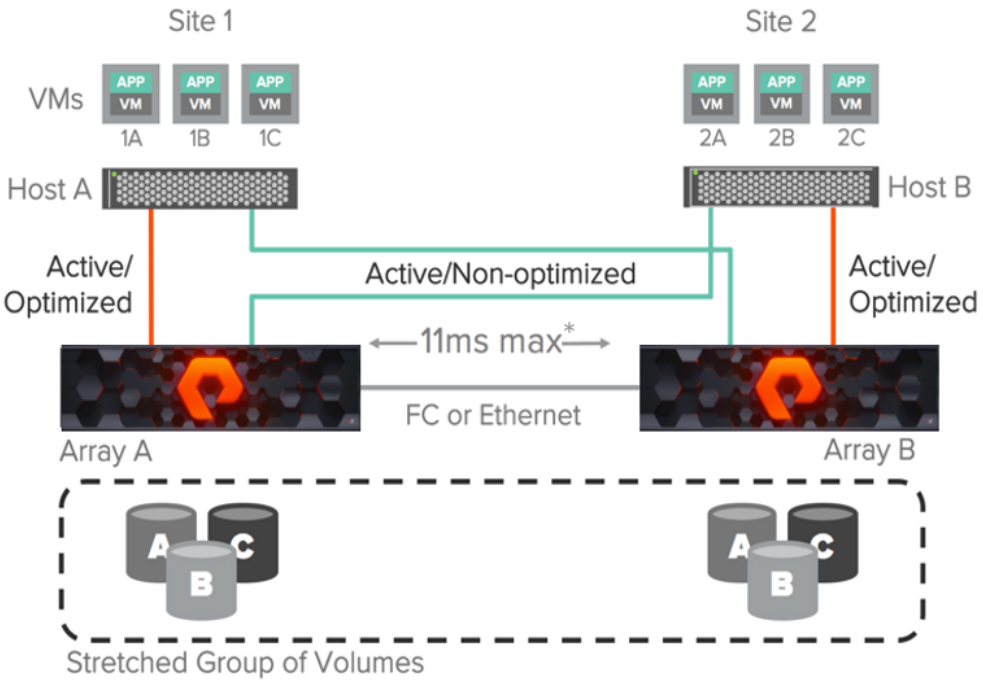


*SUBJECT TO PROTOCOL LIMITS. SEE THE Purity Replication Requirements KB FOR THE MOST CURRENT INFORMATION.

In other metro storage clustering (MSC) solutions, the continuous management of this difference in latency can be a challenge. Most Non-Pure MSC solutions make use of per-volume Asymmetric Logical Unit Access (ALUA) which is a mechanism that allows a storage array to advertise path priorities to a host. The host can then distribute it's I/O on optimized paths and avoid sending I/O on non-optimized paths. When using other non-Pure solutions storage administrators must take care to make sure VMs run only on hosts that have local optimized access to the volume in which that VMs data exists. Performing a migration of a VM to a host in the other site can cause that VM to experience 2X the latency between the sites for each write it performs if the new host does not have local optimized access to that volume, as described earlier. To remedy this the VM's data must be migrated to a different volume that allows local optimized access to for the VM's new host. This is because other non-Pure MSC solutions manage optimized paths on a per volume basis where each volume can only have optimized paths on one array or the other, but not both at the same time.

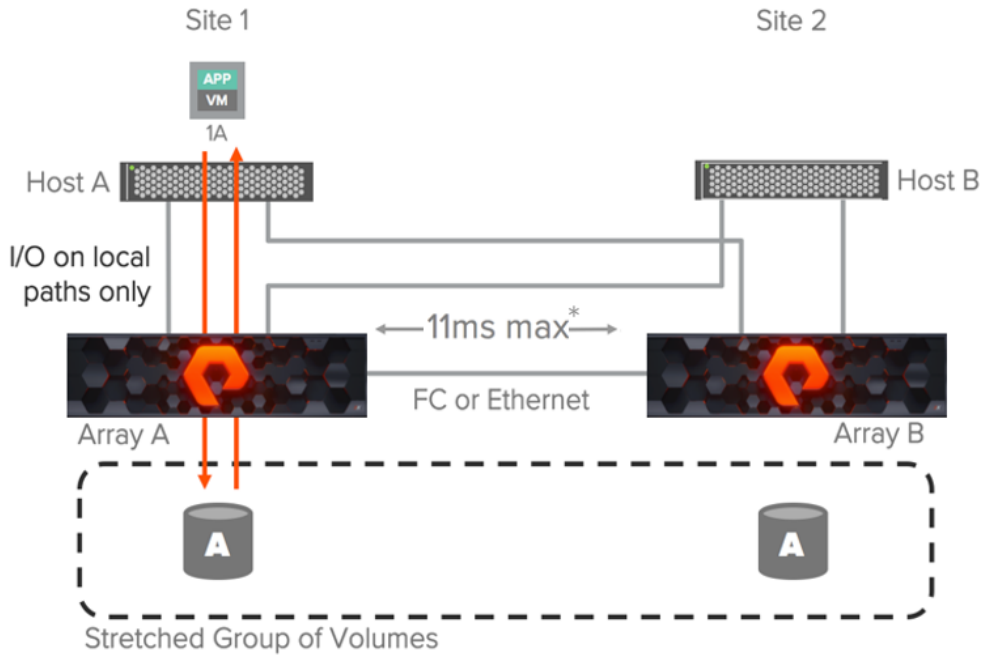
With Pure Storage Purity ActiveCluster there are no such management headaches. ActiveCluster does make use of ALUA protocol to expose paths to local hosts as active/optimized paths and expose paths to remote hosts as active/non-optimized. However, there are two advantages in the ActiveCluster implementation.

1. In ActiveCluster volumes in stretched pods are read/write on both arrays. There is no such thing as a passive volume that cannot service both reads and writes.
2. The optimized path is defined on a per host-to-volume connection basis using a preferred-array option; this ensures that regardless of what host a VM or application is running on it will have a local optimized path to that volume.



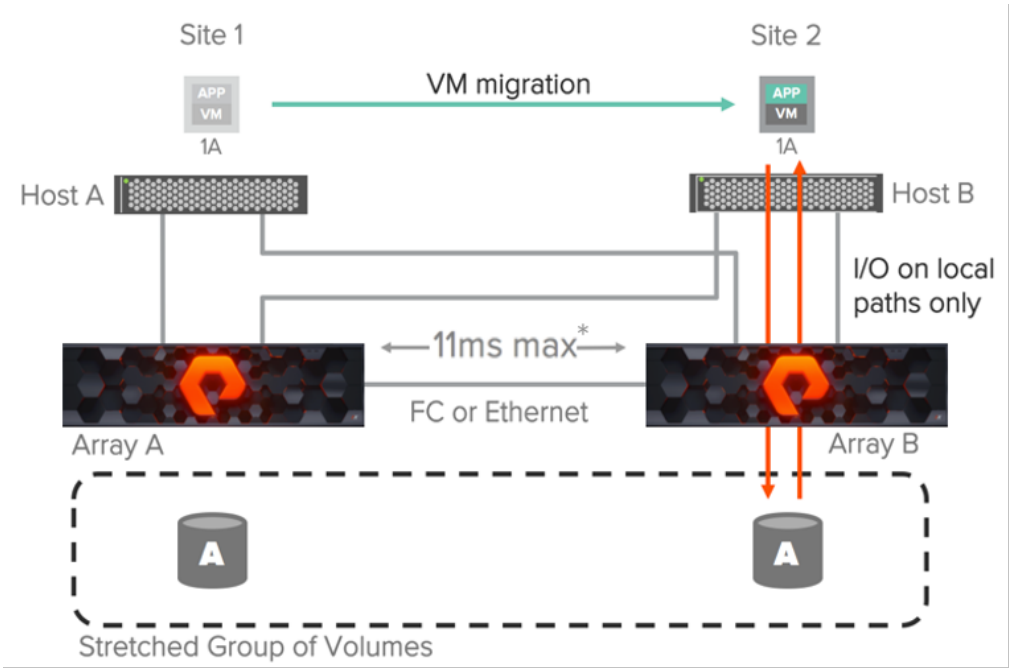
*SUBJECT TO PROTOCOL LIMITS. SEE THE Purity Replication Requirements KB FOR THE MOST CURRENT INFORMATION.

With ActiveCluster you may configure truly active/active datacenters and do not have to care what site or host a VM runs on; the VM will always have the same performance regardless of site. While a VM 1A is running on host A accessing volume A it will use only the local optimized paths as shown in the next image.



*SUBJECT TO PROTOCOL LIMITS. SEE THE Purity Replication Requirements KB FOR THE MOST CURRENT INFORMATION.

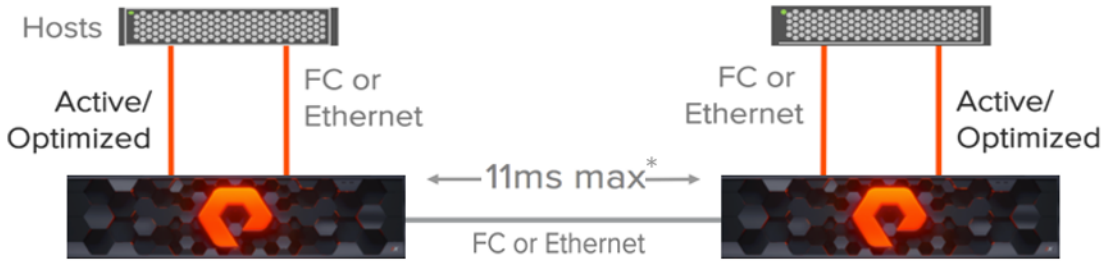
If the VM or application is switched to a host in the other site, with the data left in place, only local paths will be used in the other site as shown in the next image. There is no need to adjust path priorities or migrate the data to a different volume to ensure local optimized access.



*SUBJECT TO PROTOCOL LIMITS. SEE THE Purity Replication Requirements KB FOR THE MOST CURRENT INFORMATION.

Non-Uniform Storage Access

A non-uniform storage access model is used in environments where there is host-to-array connectivity of either FC or ethernet (for iSCSI) only locally within the same site. Ethernet connectivity for the array-to-array replication interconnect must still exist between the two sites. When deployed in this way each host has access to a volume only through the local array and not the remote array. The solution supports connecting arrays with up to 11ms of round-trip time (RTT) latency between the arrays.



*SUBJECT TO PROTOCOL LIMITS. SEE THE Purity Replication Requirements KB FOR THE MOST CURRENT INFORMATION.

Hosts will distribute I/Os across all paths to the storage only, because only the local Active/Optimized paths are available.

Comparing Uniform and Non-Uniform Storage Access

Uniform storage access may be said to provide greater application availability because it allows for applications to simply continue running without a restart in the event of a single array outage.

A multipath enabled host that is connected for uniform storage access will distribute its IOs across the local active/optimized paths according to its path selection policy: either round robin (RR) or least queue depth (LQD). If no active/optimized paths are available, the multipathing software in the host will use the active/non-optimized paths to continue access to the storage. In a uniform access connected environment, if an array outage occurs, the applications may simply continue to run with no more than a short pause in I/O as the storage access fails over from the local paths to the remote paths to the other array. This is similar to the way storage access fails over between two controllers in a single array during non-disruptive hardware and software upgrades. There is no application outage or VM restart incurred.

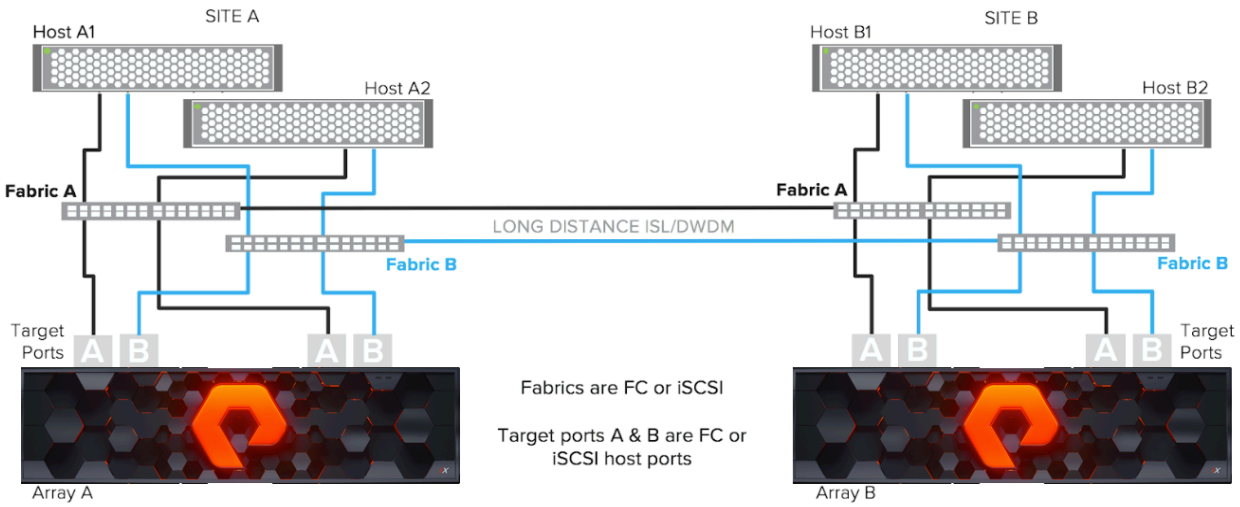
A host that is connected for non-uniform storage access will distribute its IOs across the local active/optimized paths according to its path selection policy however a host has no access to the remote array. In a non-uniform configuration, in the event of a single array outage, any applications or VMs using the affected array must be restarted on hosts in the other site that still have access to the local storage there. This restart may be automatic, depending on the type of clustering software deployed in the host layer.

Environment Connectivity

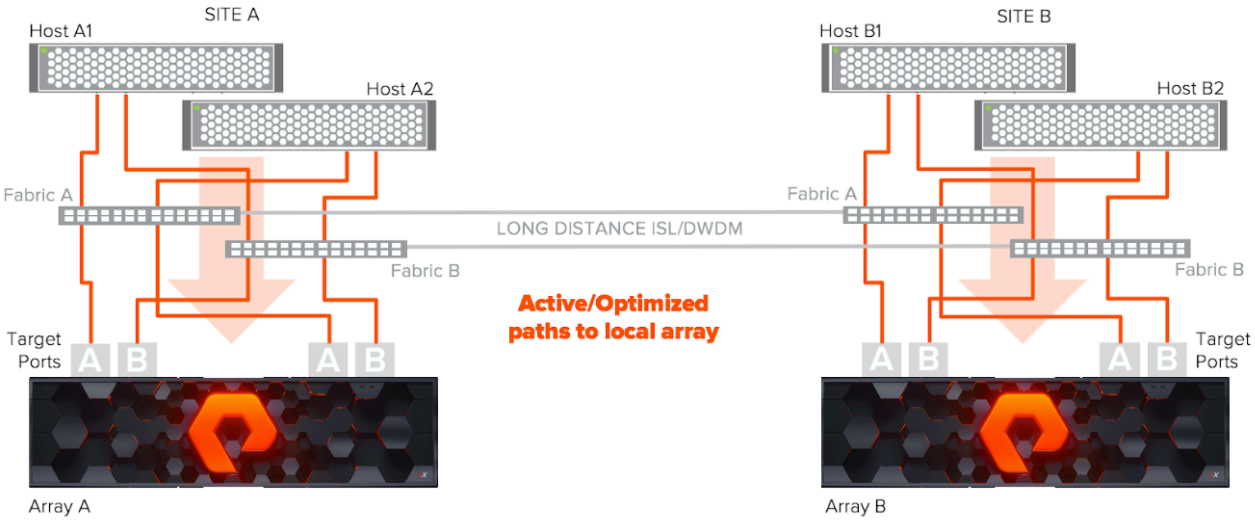
The following section describes basic connectivity examples for array-to-host connections and array-to-array connections.

Uniform Storage Connectivity

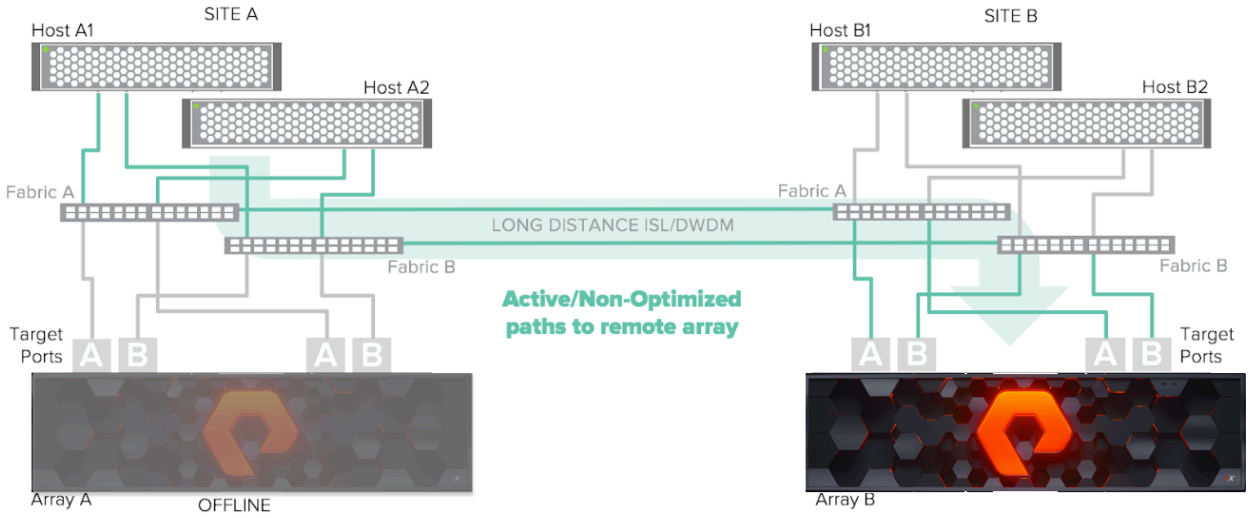
A typical uniform storage access configuration consists of two sites, with hosts at each site connected to a stretched SAN for host-to-array connectivity. The example below shows a simple dual stretch fabric configuration with the minimum number of array ports required for redundancy across the fabrics and sites: two ports per fabric, one on each controller. There are two fabrics, A and B (either FC or iSCSI) with the use of long distance inter switch links (ISLs), long haul fiber, dark fiber, DWDM, or other long-distance connectivity options providing site to site connectivity.



Host I/O to storage will remain constrained within each site according to the ALUA priorities and the preferred-array connection setting described earlier, while both arrays are online. All front-end ports on the FlashArrays are active for local hosts.



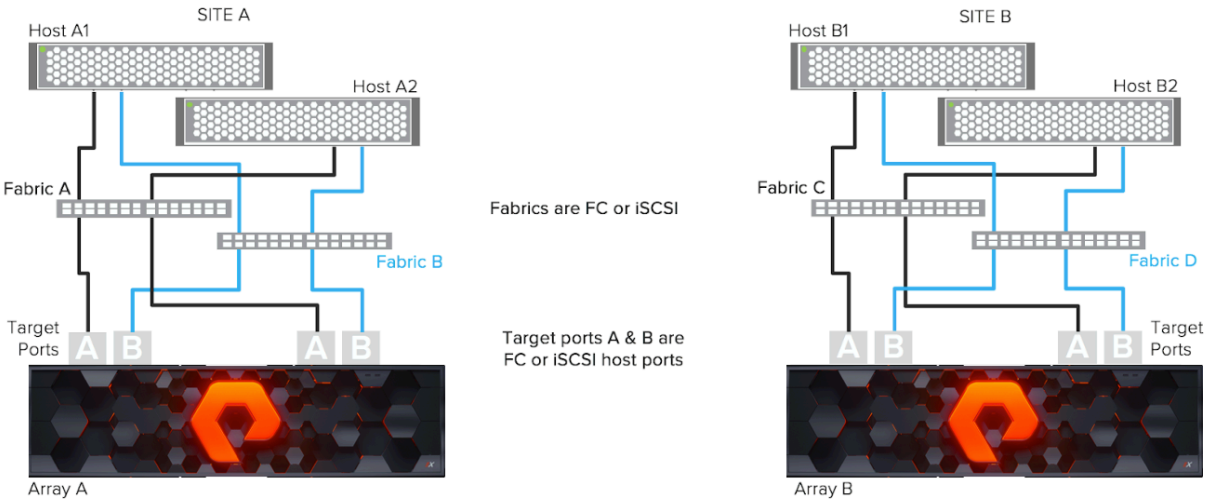
If an array in either site goes offline, for example array A, hosts in site A will continue access to storage over the non-optimized paths to array B. The same applies for the hosts in site B if the array in site A were to go offline.



Non-Uniform Storage Connectivity

A typical non-uniform storage access configuration consists of two sites, with hosts connected via single or dual fabrics within each site for host-to-array connectivity. The example below shows a simple dual

fabric configuration with the minimum number of array ports required for redundancy across the fabrics: two per fabric, one on each controller. There are two fabrics in each site, A and B (either FC or iSCSI) in site A and fabrics C and D in site B.



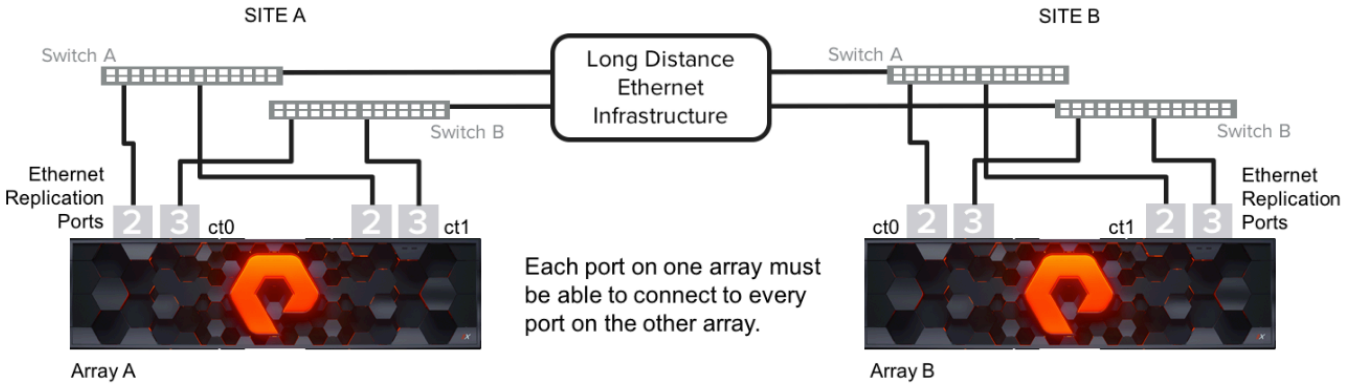
Host I/O occurs only on the local paths to each array and all paths on each array are used as active/optimized paths by the local hosts. As discussed earlier, applications or VMs must be restarted in the other site if an array goes offline.

Replication Network Connectivity

ActiveCluster supports either IP or FC based replication network technology. The following sections show the replication network connectivity options for ActiveCluster.

IP Replication Network Connectivity

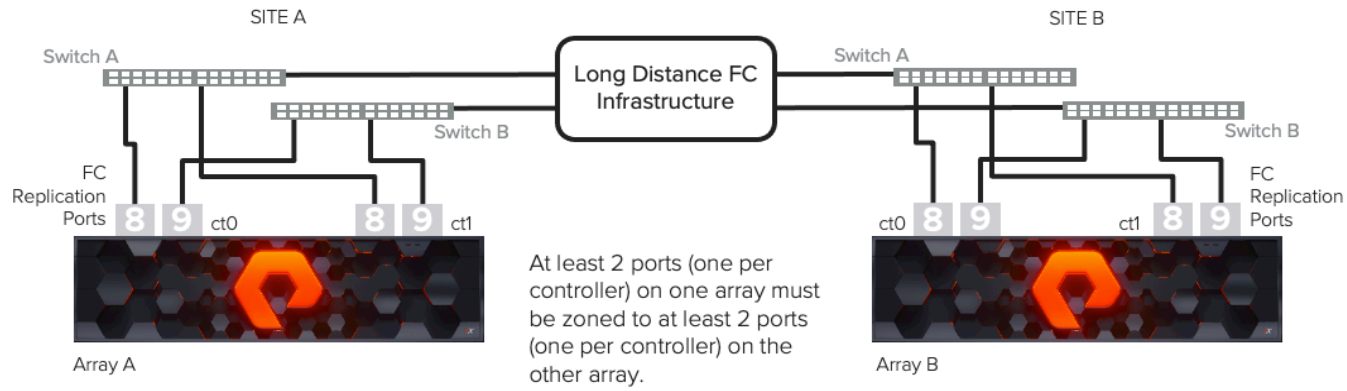
Replication connectivity for synchronous replication over IP requires two ethernet ports per controller that must be connected via a switched infrastructure such that every replication port on one array is able to connect to every replication port on the other array as services may move from port to port depending on local failover events in each array. For redundant configurations using dual switches, each controller must have a connection to each local switch and the switching infrastructure must still allow all replication ports to be able to connect to each other.



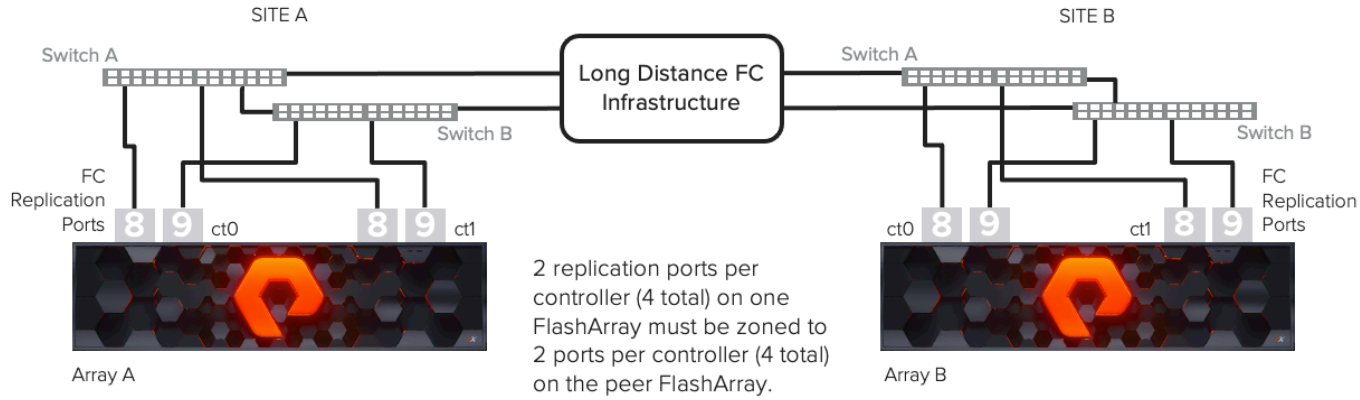
Fibre Channel Replication Network Connectivity

Replication connectivity for synchronous replication over FC requires two ports per controller, four total per FlashArray. The FC ports must be connected via a switched SAN such that at least one FC replication port on each controller is able to connect to at least one FC replication port on the each of the peer FlashArray controllers. FC connectivity between all peer controllers is required as services may move from FC port to FC port depending on local failover events in each array. With redundant switch configurations, each controller must have a FC connection to each local switch and the SAN must allow each FC replication port to be able to connect to at least one FC port on each peer FlashArray controller.

Dual Fabric Topology



Single Fabric Topology



Management Network Connectivity

Both arrays must be able to connect to each other via their management ports in order to establish a replication connection.

In the event that a failover is required, the connection to the mediator occurs from the controller management ports. So, each controller must have two management ports configured that are capable of accessing the mediator. This provides management port redundancy.

ActiveCluster Features

Transparent Failover

The concept of a storage failover does not apply to ActiveCluster in the traditional sense. ActiveCluster is designed to be truly active/active with no notion of volumes being active on one array but passive on the other. Either array can maintain I/O service to synchronously replicated volumes.

Deployments using a uniform storage access configuration essentially have failover-less maintenance of operations. In the event of an array failure, or a replication link failure causing one array to stop I/O service, the hosts experience only the loss of some storage paths but continue using other paths to the available array. There is no storage failover process for the storage administrator to execute.

In a non-uniform storage access configuration, applications or VMs running on hosts connected to the array that has stopped I/O service will need to be restarted or moved to hosts connected to the other array. Failover of the applications or VMs is managed at the host layer with application or hypervisor clustering, this may be automatic depending on the host configuration. Again, there is no storage failover process for the storage administrator to execute.

How Transparent Failover Occurs

Transparent failover between arrays in ActiveCluster is automatic and requires no intervention from the storage administrator. Failover occurs within standard host I/O timeouts similar to the way failover occurs between two controllers in one array during non-disruptive hardware or software upgrades.

ActiveCluster is designed to provide maximum availability across symmetric active/active storage arrays while preventing a split-brain condition from occurring. Split brain being the case where two arrays might serve I/O to the same volume, without keeping the data in sync between the two arrays.

Any active/active synchronous replication solution designed to provide continuous availability across two different sites requires some sort of quorum mechanism. These are sometimes referred to as a witness, a tie breaker service, voting disks, or a mediator. As their names imply, these services help each site determine the appropriate failover response and avoid split brain scenarios. ActiveCluster includes a simple to use, lightweight, and automatic way for applications to transparently failover, or simply move between sites in the event of a failure without user intervention: The Pure1 Cloud Mediator.

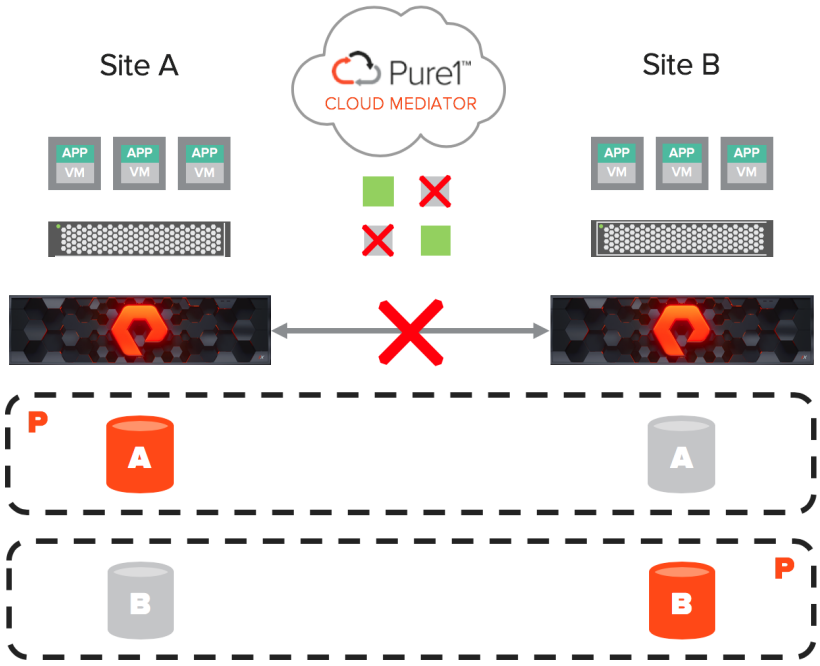
The Pure1 Cloud Mediator is responsible for ensuring that only one array is allowed to stay active for each pod when there is a loss of communication between the arrays.

In the event that the arrays can no longer communicate with each other over the replication interconnect, both arrays will pause I/O and reach out to the mediator to determine which array can stay active for each sync replicated pod. The first array to reach the mediator is allowed to keep its synchronously replicated pods online. The second array to reach the mediator must stop servicing I/O to its synchronously replicated volumes, in order to prevent split brain. The entire operation occurs within standard host I/O timeouts to ensure that applications experience no more than a pause and resume of I/O.

Failover Preferences

Active-Active datacenters often have workloads that tend to run at one site or at the other. The site that applications tend to run in may be determined by historical convention, administrative convenience, infrastructure disparity, primary user location, or any of a number of other reasons. When ActiveCluster performs a mediator race following the loss of the synchronous replication links between datacenters, the outcome of the resulting race to the mediator can be unpredictable. For non-uniform host connectivity, the lack of mediator race predictability can mean a disruptive restart for applications running on stretched pod volumes in the losing FlashArray.

ActiveCluster provides a failover preference feature which allows the mediator race to be administratively influenced on a per pod basis. This capability enables administrators align pod failover behavior with the site where each application tends to run. Functionally, the failover preference feature gives the preferred FlashArray for each pod a 6 second head start in its race to the mediator.



In the illustration above, the stretched pod containing volume A prefers Site A (preference is indicated by the orange letter P at the top left corner of the A pod). Due to the head start it is given, Pod A is more likely to win its race to the mediator and stay online at Site A. The second pod with volume B prefers site B (preference is indicated by the orange letter P at the top right corner of the B pod). Pod B is more likely to stay online at Site B after winning its race to the mediator. In this way, site and array alignment with applications can be established allowing hosts that use non-uniform connectivity to continue without the need for disruptive restart.

As described earlier, non-uniform host connectivity hosts are connected to just one FlashArray. If that one array suspends stretched pod volumes, the applications running on them must be disruptively restarted by host cluster software on hosts that have connections to the remote FlashArray. By setting a failover preference for Site A or at Site B, the ActiveCluster administrator can mitigate against a disruptive restart of his applications. Disruptive restarts will be confined to cases where one FlashArray is offline, unreachable, or if there is an entire site loss. Setting a failover preference for pods supporting clustered applications running on non-uniformly connected hosts is, therefore, a recommended best practice.

The key advantage with using a failover preference setting compared to allowing a fixed site bias is that if the preferred array fails (or if the site is lost), the non-preferred array can still win the race to the mediator and keep all pod volumes online. A static, fixed, bias can lead to a total cluster outage as the non-preferred array must suspend IO regardless of what happens to the preferred array.

Pre-Election

Starting with Purity 5.3, ActiveCluster uses its built-in Mediator polling mechanism to allow both FlashArrays to jointly agree a mediation race winner for each stretched pod if **both** arrays are unable to

reach the mediator. Pre-Election also makes use of the pod failover preference setting (if set) to determine the winning FlashArray for each pod. If no pod failover preference was set, a winner will be selected automatically.

> ***Note:*** *The Mediator polling cycle on each FlashArray is independent. As a result, the maximum amount of time to identify the loss or the return of the Mediator by* ***both*** *arrays can be up to* ***5 minutes****.*

This means that if the Pure1 Cloud Mediator (or the On-Premises Mediator) is offline or unreachable for an extended period of time, ActiveCluster can still provide access to stretched pod volumes if the replication links should subsequently fail.

Pre-Election automatically disengages when one or both of the FlashArrays re-establishes contact with the Mediator. Pre-Election is always on and it requires no administrative setup, no ongoing monitoring, and no upgrade to the Mediator (cloud or on-prem). See the **Access to Storage Through Failures** section of this document for more details around specific failure outcomes.

The Pure1 Cloud Mediator

A failover mediator must be located in a 3rd site that is in a separate failure domain from either site where the arrays are located. Each array site must have independent network connectivity to the mediator such that a single network outage does not prevent both arrays from accessing the mediator. A mediator should also provide a very lightweight and easy to administer component of the solution. The Pure Storage solution provides this automatically by utilizing an integrated cloud-based mediator. The Pure1 Cloud Mediator provides two main functions:

1. Prevent a split-brain condition from occurring where both arrays are independently allowing access to data without synchronization between arrays.
2. Determine which array will continue to service IO to synchronously replicated volumes in the event of an array failure, replication link outage, or site outage.

The Pure1 Cloud Mediator has the following advantages over a typical non-Pure heavy-handed voter or witness component:

- **SaaS operational benefits** - As with any SaaS solution the operational maintenance complexity is removed: nothing to install onsite, no hardware or software to maintain, nothing to configure and support for HA, no security patch updates, etc.
- **Automatically a 3rd site** - The Pure1 Cloud Mediator is inherently in a separate failure domain from either of the two arrays.
- **Automatic configuration** - Arrays configured for synchronous replication will automatically connect to and use the Pure1 Cloud Mediator.
- **No misconfiguration** - With automatic and default configuration there is no risk that the mediator could be incorrectly configured.

- **No human intervention** - A significant number of issues in non-Pure active/active synchronous replication solutions, particularly those related to accidental split brain, are related to human error. Pure's automated non-human mediator eliminates operator error from the equation.
- **Passive mediation** - Continuous access to the mediator is not required for normal operations. The arrays will maintain a heartbeat with the mediator, however if the arrays lose connection to the mediator they will continue to synchronously replicate and serve data as long as the replication link is active.

Advantages of a Passive Mediator

In **non-Pure FlashArray** metro storage clustering (MSC) solutions, active voter or witness components are used that use a heavy-handed approach to drive a failover. These types of voters or witnesses introduce the risk of split brain. An active voter or witness is an automation tool that directly sends commands to one array initiating a failover in the event that the other array becomes isolated. In non-Pure solutions an active voter or witness risks split brain because even though a site is isolated, it could still be servicing out-of-sync data to applications internally in the isolated site. Short interruptions of connectivity to an active voter or witness can lead to unnecessary failovers. Should an intermittently isolated site come back online after an active mediator has caused a failover, split brain will occur.

The Pure1 Cloud Mediator is a passive mediator. The mediator does not actively execute the failover of the arrays and does not perform as a quorum voter, it is simply used by the arrays to determine or authorize what action the array should take in the event that the ability to synchronously replicate is lost. Access to the mediator is required in the event of a failure of an array, failure of the replication link, or a site failure; the mediator is then required to ensure one array will continue to serve data for volumes in stretched pods. Continuous access to the mediator is not required. Access to the mediator is monitored by the arrays and an alert will be generated if the mediator is not available.

On-Premises Failover Mediation

Failover mediation for ActiveCluster can also be provided using an on-premises mediator distributed as an OVF file and deployed as a VM. Failover behaviors are exactly the same as described above. The on-premises mediator simply replaces the role of the Pure1 Cloud Mediator during failover events.

The on-premises mediator has the following basic requirements:

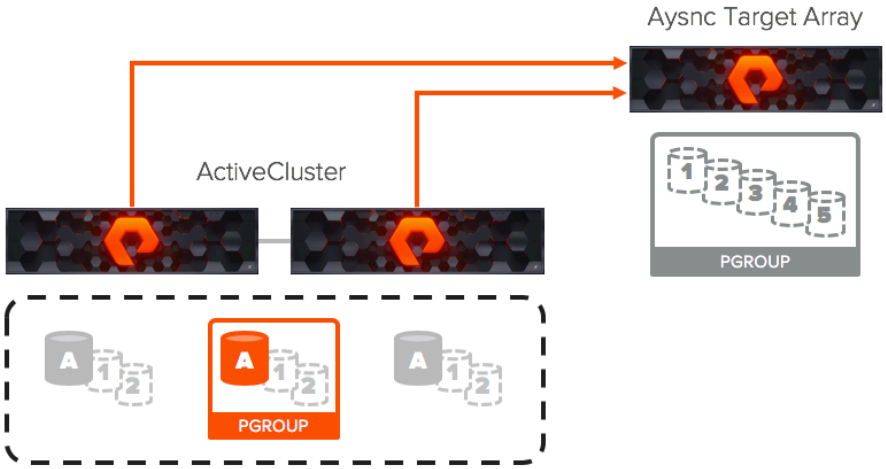
- The on-premises mediator can only be deployed as a VM on virtualized hardware. It is not installable as a stand-alone application.
- High Availability for the mediator must be provided by the hosts on which the mediator is deployed. For example, using VMware HA, or Microsoft Hyper-V HA Clustering.
- Storage for the mediator must not allow the configuration of the mediator to be rolled back to previous versions. This applies to situations such as storage snapshot restores, or cases where the mediator might be stored on mirrored storage.
- The arrays must be configured to use the on-premises mediator rather than the Pure1 Cloud Mediator.
- The mediator must be deployed in a third site, in a separate failure domain that will not be affected by any failures in either of the sites where the arrays are installed.

- Both array sites must have independent network connections to the mediator such that a failure of one network connection does not prevent both arrays from accessing the mediator.

Active-Active Asynchronous Replication

Active-Active Asynchronous provides the highest levels of Pure FlashArray data protection. Along with the easy to use, space efficient, and bandwidth friendly features you've come to expect, it also provides a more robust, target orchestrated, and integrated way to protect ActiveCluster volumes. Asynchronous replication automatically continues during source array failure without administrative intervention or the need to completely re-copy (aka re-baseline) data. Replication content is intelligently requested by the Target array to optimize and load-balance each replication link to the source arrays. By leveraging this feature, ActiveCluster volumes are seamlessly replicated to a 3rd site which provides an additional level of protection for your data even when disaster impacts multiple, sync distance, metropolitan areas.

To better understand the benefits of the Active-Active Asynchronous feature, let's take a brief look at how it works.

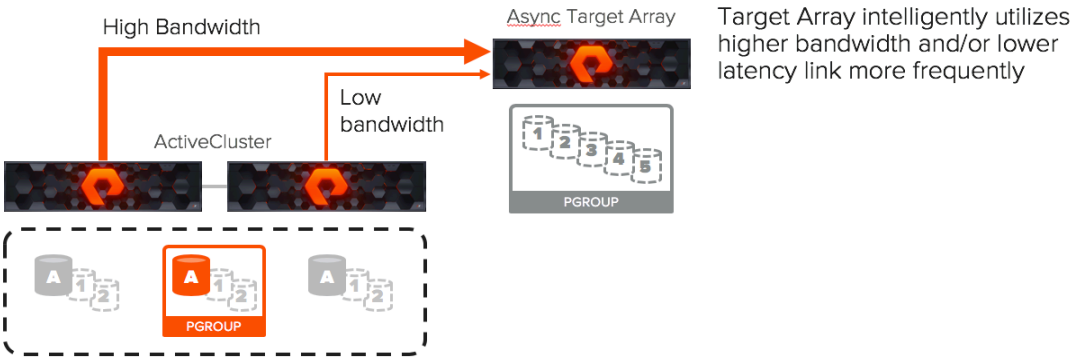


In the illustration above, ActiveCluster volumes are organized and managed via Pods (indicated by the dashed black lines). Within Pods, volumes that are also asynchronously replicated are grouped into Protection Groups (indicated by the PGroup with solid orange lines). Snapshots of each PGroup and the volumes they contain (indicated by the grey volumes with dashed lines) are replicated the target array and retained based on an administratively defined per-PGroup schedule.

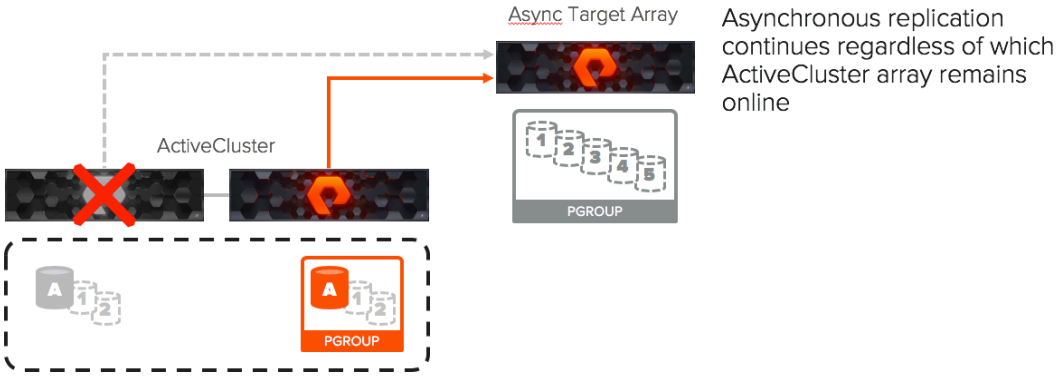
The Async Target Array requests different portions of unique content from each of the source ActiveCluster arrays. In other words, the same content request is **not** sent to both ActiveCluster arrays. The Target Array does not request content that it already has even it if is contained within the snapshot residing on the ActiveCluster Arrays. Content requests are sent based on which source array returns the content quickest so that differences in latency and bandwidth are optimized to provide best overall replication speed and efficiency.

Active-Active Asynchronous Replication Advantages

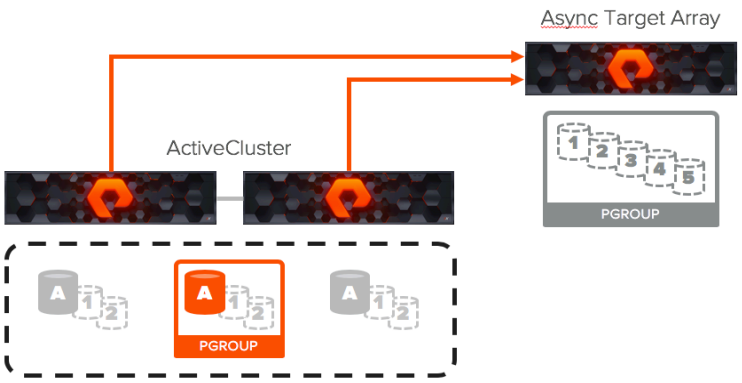
- **INTELLIGENT REPLICATION TECHNOLOGY**: Patent-pending target-orchestrated asynchronous replication intelligently load-balances discrete content requests between the ActiveCluster FlashArrays. ActiveCluster FlashArrays that complete content requests the quickest are identified by the Target FlashArray and will be more frequently utilized, which optimizes overall link utilization.



- **NO RE-BASELINE**: The loss of either source array or a replication link is transparent to Asynchronous replication and requires no re-baseline allowing replication to Target Array to be maintained automatically.



- **AUTOMATIC FAILOVER, LOAD BALANCING, RECOVERY**: Following link and/or array restoration, replication automatically resumes pulling snapshot content from both source arrays.



Async replication automatically resumes transfer from both sites after recovery

- **PROTECTION FROM REGIONAL DISASTER**: 3rd-site out of region asynchronous data replication to anywhere on the globe
- **NO BOLT-ONS & NO LICENSES**: Install or Upgrade to Purity 5.2 or higher, and you're ready to configure Active-Active Asynchronous Replication
- **SIMPLE MANAGEMENT**: Managed via native FlashArray API / CLI / GUI from either source array
- **BEST OF BREED TECHNOLOGY**: Competing solutions don't offer the simplicity, capability, or resiliency of the Purity Active-Active Async feature. These solutions are most often add-on products with limited integration and separate administrative interfaces. They also commonly require additional hardware or appliances, dedicated storage capacity, more rack space, more SAN ports, additional licensing, and professional services for installation.

ActiveCluster Failover Scenarios

As discussed earlier, the concept of a storage failover does not apply to Pure Storage ActiveCluster in the traditional sense. ActiveCluster provides Transparent Failover which is an event where I/O continues to volumes in a stretched pod at one FlashArray but stops to the same stretched pod volumes at the other FlashArray. In doing so, I/O transparently continues without impact to mission critical application I/O.

The behavior of the environment during some failure events differs depending on the host access configuration, uniform or non-uniform. In a uniform storage access configuration hosts may simply experience the loss of some storage paths; there is no storage failover process. In a non-uniform storage access configuration, there is again no storage failover process, however the applications or VMs running on hosts connected to the offline array will have to be moved to other hosts. That may happen automatically by application or hypervisor cluster software in the host.

Any volumes that are not in stretched pods at the time of such a failure will remain online as normal if the array remains online.

How to Failover

Failover is automatic, and no storage administrator intervention is necessary to activate a failover with ActiveCluster. An automatic failover requires at least one array with access to the mediator and may be triggered by any of the following:

- Failure of an array.
- Failure of the replication link between the two arrays.
- Failure of an entire array site, a site wide disaster.

A complete failure of the storage network in one site may be said to be a failover but it is not a storage failover. In such a case both arrays are able to service I/O, but access is only occurring for hosts in the site where the storage network is still online.

How to Resynchronize and Recover

Resynchronization and recovery are automatic, and no storage administrator intervention is necessary to resynchronize and recover replication with ActiveCluster.

Should the two arrays in a pod go out of sync, internal checkpoints that are created periodically will provide a known in-sync state from which the arrays can automatically resynchronize. When the link is restored any changes made since the outage on the array that kept the stretched pod online will be asynchronously transferred to the other array. The arrays will get in sync via shorter and shorter periodic asynchronous background transfers. Once the arrays are nearly synchronized, they will smoothly transition to synchronous replication mode and the host paths to the offline side of the pod will be automatically restored, allowing the hosts to perform I/O through both arrays again.

Note: The Active-Active Asynchronous replication provides out-of-region copies of data protected by ActiveCluster on a third FlashArray. These copies can be used to restore data to the ActiveCluster arrays in the event **both** ActiveCluster FlashArrays are lost or experience data loss.

Access to Storage Through Failures

The table below describes whether or not ActiveCluster is able to service I/O for volumes that are configured for synchronous replication, on one array or the other, when a component outage occurs. Prior to Purity 5.3, 2 of the 3 main solution components (FlashArray A, FlashArray B, mediator) must be online for IO service to continue on the surviving ActiveCluster FlashArray. Starting with Purity 5.3, the Pre-Election feature can enable a single AC FA to keep stretched pod volumes online in scenarios where Pre-Election has completed ahead of replication links failing or the failure of the non-Pre-Elected FlashArray.

Asynchronous target FlashArray storage access is not impacted by the loss of either or both source ActiveCluster FlashArrays or due to replication link failures.

Replication link failure or array failure, while the mediator is unavailable, results in stop of I/O to sync replicated volumes on both arrays to prevent split brain. The mediator is a required component of ActiveCluster.

Solution Component Failure				Access to Storage Through Purity 5.2	Access to Storage Starting with Purity 5.3
One Array	Other Array	Replication Link	Mediator		
UP	UP	UP	UP	Available on both arrays	Available on both arrays
UP	DOWN	UP	UP	Available on one array	Available on one array
UP	UP	DOWN	UP	Available on one array	Available on one array
UP	UP	UP	DOWN	Available on both arrays	Available on both arrays
UP	DOWN	DOWN	UP	Available on one array	Available on one array
UP	UP	DOWN*	DOWN*	Unavailable	Unavailable
UP	DOWN*	UP	DOWN*	Unavailable	Unavailable
UP(pre-elected)	UP	DOWN	DOWN**	Unavailable	Available on one array
UP(pre-elected)	DOWN	UP	DOWN**	Unavailable	Available on one array***

* Simultaneous failures of components

** Pre-Election completes before second component failure.

*** Assumes the 'Other Array' was not the Pre-Elected array. If the Pre-Elected array fails, then the stretched pod volumes are unavailable.

Note: If the Mediator becomes unavailable after an array failure or a replication link failure has already been sustained, access to the mediator is no longer required and access to storage remains available on one array.

Failure Scenario Behaviors

The following table describes the effect of various failure events in the environment. In this table the following terminology is used.

Mediator - Refers to the Pure1 Cloud Mediator or the on premises VM mediator. In either case the mediator is located at a third site outside of the failure domain of either array's site. The failover behaviors are the same regardless of which mediator is used.

Mediation winner - The array that is first to reach the mediator and therefore keep data services online for volume in stretched pods.

Mediation loser - The array that is unable to reach the mediator, or is 2nd to reach the mediator, and therefore must turn data services offline for volumes in stretched pods.

Array, Replication Network, and Site Failures

Failure Scenario	Failure Behavior
Local HA controller failover in one array	After a short pause for the duration of the local HA failover, host I/O will continue to both arrays without losing RPO-Zero Active-Active Asynchronous Replication resumes automatically
Replication link failure	Each Pod: • experiences a short pause, host IO continues to volumes only on the array that contacts the mediator first. • fails over automatically and transparently, with no administrative intervention. Uniform connected hosts: • after a short pause in IO, continue IO to the array that won the race to the mediator. • experience some storage path failures for paths to the array that lost the race to the mediator. • in the mediator losing site will maintain access to volumes remotely across stretched SAN to the mediator winning site. Non-uniform connected hosts: • in the mediator winning site will maintain access to volumes with no more than a pause in IO. • in the mediator losing site will experience total loss of access to volumes. • use host cluster software to recover the apps to a host in the mediator winning site. This may be automatic depending on the type of cluster. Active-Active Asynchronous Replication: • Failover is automatic and transparent, and no administrative intervention is necessary. • Async replication to Target Array is maintained automatically.
Loss of contact with Mediator	No effect. Host IO continues through all paths on both arrays as normal
Entire single array failure	After a short pause, Host IO automatically continues on surviving array.

	Failover is automatic and transparent, and no administrator intervention is possible or necessary. Uniform connected hosts: • in the surviving array site, after a short pause in IO, continue IO to the surviving array that was able to reach the mediator. • experience some storage path failures for paths to the failed array. • in the site where the array failed will do IO to volumes remotely across the stretched SAN to the surviving array. Non-uniform connected hosts: • in the surviving array site, after a short pause in IO, continue IO to the surviving array that was able to reach the mediator. • in the failed array site will experience total loss of access to volumes. • use host cluster software to recover the apps to a host in the other site. This may be automatic depending on the type of cluster. Active-Active Asynchronous Replication: • Failover is automatic and transparent, and no administrator intervention is necessary. • Async replication to Target Array is maintained automatically.
Entire site failure	After a short pause, Host IO automatically continues on surviving array. Failover of the array is automatic & transparent, and no administrator intervention is possible or necessary. Uniform connected hosts: • in the surviving array site, after a short pause in IO, continue IO to the surviving array that was able to reach the mediator. • experience some storage path failures for paths to the array in the failed site. • use host cluster software to recover the apps to hosts in the surviving site. This may be automatic depending on the type of cluster. Non-uniform connected hosts: • in the surviving site, after a short pause in IO, continue IO to the surviving array that was able to reach the mediator. • in the surviving site will maintain access to local volumes with no more than a pause in IO. • use host cluster software to recover the apps to hosts in the surviving site. This may be automatic depending on the type of cluster. Active-Active Asynchronous Replication: • Failover is automatic and transparent, and no administrator intervention is necessary. • Async replication to Target Array is maintained automatically.

Prior to Purity 5.3 Mediator failure **followed by a second failure** of replication link, or failure of one array, or failure of one site. (2nd failure occurs while mediator is unavailable)	Host IO access is lost to sync rep volumes on both arrays. This is a double failure scenario; data service is not maintained through failure of either array if the mediator is unavailable. Options to recover: 1. Restore access to either the mediator or replication interconnect and the volumes will automatically come back online, as per above scenarios. OR 2. Clone a pod create new volumes with different LUN serial numbers. New LUN serial numbers will prevent hosts from automatically connecting to and using the volumes, avoiding split brain. Then re-identify and reconnect all LUNs on all hosts.
Purity 5.3 and later Mediator failure **detected** by both FlashArrays **followed by a second failure** of replication link, or failure of one array, or failure of one site. For Mediator failures that coincide or are immediately followed by a second failure of the replication links, a failure of one array, or a site failure, the host IO access behavior is the same as pre-Purity 5.3 versions.	Host IO access continues at Pre-Elected FlashArray for all replication link failures. Host IO access continues at Pre-Elected FlashArray provided the Pre-Elected array is not the array that failed and is not located in the site that failed. Host IO access at non-Pre-Elected array will remain suspended until a connection to the Pre-Elected (winning) array is re-established. Options to recover stretched volume data on non-Pre-Elected array: 1. Restore replication interconnect to Pre-Elected (winning array) and the volumes will automatically come back online, as per above scenarios. OR 2. Clone a pod create new volumes with different LUN serial numbers. New LUN serial numbers will prevent hosts from automatically connecting to and using the volumes, avoiding split brain. Then re-identify and reconnect all LUNs on all hosts.

Host and Storage Network Failures

Failure Scenario	Failure Behavior
Single or multiple host failure	Applications can automatically failover to other hosts in the same site or other hosts in the other site connected to the other array. This is driven by host cluster software assuming clusters are stretched between sites. VMware HA, Oracle RAC, SQL Cluster, etc.
Stretched SAN fabric outage (FC or iSCSI) (failure of SAN interconnect between sites)	Host IO automatically continues on local paths in the local site. Uniform connected hosts: • experience some storage path failures for paths to the remote array and continue IO on paths to the local array. • in each site will maintain access to local volumes with no more than a pause in IO. Non-uniform connected hosts: • do not have a SAN interconnect between sites, so this scenario is not applicable.
SAN fabric outage in one site	Applications can automatically failover to hosts at the other site connected to the other array.

	This is driven by host cluster software assuming clusters are stretched between sites. VMware HA, Oracle RAC, SQL Cluster, etc. Uniform connected hosts: • in the site without the SAN outage, experience some storage path failures for paths to the remote array and continue IO on paths to the local array. • in the site with the SAN outage, will experience total loss of access to volumes and applications must failover to the other site as mentioned above. Non-uniform connected hosts: • in the site without the SAN outage, will maintain access to local volumes. • in the site with the SAN outage, will experience total loss of access to volumes and applications must failover to the other site as mentioned above.

Requirements

The following items are required to use Purity ActiveCluster.

Solution Component Requirements

- Two Pure Storage FlashArrays running Purity version 5.1.0 or higher (6.1.0 or higher for AC over FC)

  Notes:

  1. Active-Active Asynchronous configurations require Purity version 5.2.0 or higher (Purity 6.1.3 or higher w/ AC-FC) and at least 3 FlashArrays.
  2. ActiveCluster over FC requires Purity version 6.1.0 or higher

- The Pure1 Cloud Mediator or the on-premises mediator installed in a third site, in a separate failure domain from both of the array sites.
- Host HA protection for the mediator if using the on-premises mediator.

Replication Network Requirements

The replication network is used for the initial asynchronous transfer of data to stretch a pod, to synchronously transfer data and configuration information between the arrays, and to resynchronize a pod.

- Up to a maximum 11ms round trip replication network latency between FlashArrays. Maximum supported latency is protocol dependent. See Purity Replication Requirements KB for the current replication network latency limits.
- 4x IP replication ports or 4x FC replication ports per array (two per controller). Two replication ports per controller are required to ensure redundant access from the primary controller to the other array.
- For IP replication, 4x dedicated replication IP addresses per array.

- For FC replication, one or more zones enabling connectivity between the 4 dedicated FC replication ports per FlashArray
- A redundant switched replication network. Direct connecting Pure Storage FlashArrays for replication is not possible.
- Adequate bandwidth between arrays to support bi-directional synchronous writes and bandwidth for resynchronizing. This depends on the write rate of the hosts at both sites.
- Active-Active Asynchronous Replication requires one IP connection from each ActiveCluster FlashArray [**2 total**] to the third (async target) FlashArray.

Management Network Requirements

The management network is necessary to connect the arrays and to connect to the mediator. A temporary outage of the management network alone does not cause a failure.

- 4x 1GbE management ports per array (two per controller). Two management ports per controller are required to ensure redundant access from the primary controller to the mediator.
- A minimum of 5 management IP addresses per array. These are:
    - 1 IP address for vir0
    - 2 physical ethernet port IP addresses configured to support Vir0 and to be connected to the first management network.
    - 2 physical port IP addresses for connected to the second management network. These **are not** required to be configured under a virtual interface.

      Note: *For ease manageability, a second virtual interface, Vir1, can be configured to use the two IP addresses on the second management network provided a 6th, IP is available on the second management network.*
- Independent management network egress points (aka gateways) from both ActiveCluster FlashArray sites to the mediator, such that no single network outage can prevent both arrays from accessing the mediator.

  Note*: Some multi-site networks connect to the internet through just one of the two sites. This is known as a backhauled internet design. Backhauled designs are inherently a single point of failure from a Pure1 Cloud Mediator access perspective.*

  For network configurations such as these, the on-premises mediator should be used. The same requirements still apply for the on-premises mediator:
    - *Independent management network access to the mediator VM from each FlashArray*
    - *HA protection of the mediator VM*
    - *3rd fault domain (isolated 3rd site) deployment of the mediator VM*

PURESTORAGE®

Pure Storage, Inc.
Twitter: @purestorage
www.purestorage.com

650 Castro Street, Suite #260
Mountain View, CA 94041

T: 650-290-6088
F: 650-625-9667

Sales: sales@purestorage.com
Support: support@purestorage.com
Media: pr@purestorage.com
General: info@purestorage.com