Best Practices

# Dell EMC PowerFlex: Networking Best Practices and Design Considerations

PowerFlex Version 3.5.x


## Abstract

This document describes core concepts of Dell EMC PowerFlex™ software-defined storage and best practices for designing, troubleshooting, and maintaining networks for PowerFlex systems, including both single-site and multi-site deployments with replication.



April 2021

H18390.3

# Revisions

| Date | Description |
|---|---|
| April 2021 | Updates on virtual networks and dynamic routing |
| January 2021 | Inclusive language Disclaimers added |
| June 2020 | PowerFlex 3.5 release and rebranding – rewrite & updates for replication |
| May 2019 | VxFlex OS 3.0 release – additions and updates |
| July 2018 | VxFlex OS rebranding & general rewrite – add VXLAN |
| June 2016 | Add LAG coverage |
| November 2015 | Initial Document |

# Acknowledgements

Content Owner: Brian Dean, Storage Technical Marketing

Support: Neil Gerren, Igal Moshkovich, Matt Hobbs, Dan Aharoni, Rivka Matosevich

# Table of contents

# Executive summary

The Dell EMC™ PowerFlex™ family of products is powered by PowerFlex software-defined storage – a scale-out block storage service designed to deliver flexibility, elasticity, and simplicity with predictable high performance and resiliency at scale. Previously known as VxFlex OS, the PowerFlex storage software accommodates a wide variety of deployment options, with multiple OS and hypervisor capabilities.

The PowerFlex family currently consists of a rack-level and two node-level offerings: an appliance and ready nodes. This document primarily focuses on the storage virtualization software layer itself and is mostly relevant to the ready nodes, but it will be of interest to anyone wishing to understand the networking required for a successful PowerFlex-based storage system.

PowerFlex rack is a fully engineered, rack-scale system for the modern data center. In the rack solution, the networking comes pre-configured and optimized, and the design is prescribed, implemented, and maintained by PowerFlex Manager (PFxM). This document does not address the rack deployment situation. For other PowerFlex family solutions, one must design and implement an appropriate network. Starting with the release of PFxM 3.6, the appliance permits the use of unsupported commercial-grade switches, as long as they meet specific criteria and are configured to match the topology PFxM would have deployed. We cover this below.

A successful PowerFlex deployment depends on a properly designed network topology. This document provides guidance on network choices and how these relate to the traffic types among the different PowerFlex components. It covers various scenarios, including hyperconverged considerations and deployments using PowerFlex native asynchronous replication, introduced in the software version 3.5. It also covers general Ethernet considerations, network performance, dynamic IP routing, network virtualization, implementations within VMware® environments, validation methods, and monitoring recommendations.

# Audience and Usage

This document is intended for IT administrators, storage architects, and Dell Technologies™ partners and employees. It is meant to be accessible to readers who are not networking experts. However, an intermediate level understanding of IP networking is assumed.

Readers familiar with PowerFlex (VxFlex OS) may choose to skip much of the “PowerFlex Functional Overview” and “PowerFlex Software Components” sections. But attention should be paid to the new Storage Data Replicator (SDR) component.

This guide provides a minimal set of network best practices. It does not cover every networking best practice or configuration for PowerFlex. A PowerFlex technical expert may recommend more comprehensive best practices than those covered in this guide.

Cisco Nexus® switches are often used in the examples in this document, but the same principles generally apply to any network vendor.[1] For convenience, we will generally refer to any servers running at least one PowerFlex software component simply as a PowerFlex node, without distinguishing consumption options.

Specific recommendations that appear throughout in **boldface** are revisited in the “Summary of Recommendations” section at the end of this document.

[1] For some guidance in the use of Dell network equipment, see the paper on VxFlex Network Deployment Guide using Dell EMC Networking 25GbE switches and OS10EE.

# 1 PowerFlex Functional Overview

PowerFlex is storage virtualization software that creates a server and IP-based SAN from direct-attached storage to deliver flexible and scalable performance and capacity on demand. As an alternative to a traditional SAN infrastructure, PowerFlex combines diverse storage media to create virtual pools of block storage with varying performance and data services options. PowerFlex provides enterprise-grade data protection, multi-tenant capabilities, and enterprise features such as inline compression, QoS, thin provisioning, snapshots and native asynchronous replication. PowerFlex provides the following benefits:

**Massive Scalability** – PowerFlex can start with only a few nodes and scale up to many hundreds in a cluster. As devices or nodes are added, PowerFlex automatically redistributes data evenly, ensuring fully balanced pools of distributed storage.

**Extreme Performance** – Every storage media device in a PowerFlex storage pool is used to process I/O operations. This massive I/O parallelism of resources eliminates bottlenecks. Throughput and IOPS scale in direct proportion to the number of storage devices added to the storage pool. Performance and data protection optimization is automatic.

**Compelling Economics** – PowerFlex does not require a Fiber Channel fabric or dedicated components like HBAs. There are no forklift upgrades for outdated hardware. Failed or outdated components are simply removed from the system, while new components are added and data is rebalanced. In this way, PowerFlex can reduce the cost and complexity of the storage solution vs. traditional SAN.

**Unparalleled Flexibility** – PowerFlex provides flexible deployment options. In a two-layer deployment, applications and the storage software are installed on separate pools of servers. A two-layer deployment allows compute and storage teams to maintain operational autonomy. In a hyper-converged deployment, applications and storage are installed on a single, shared pool of servers, providing a low footprint and cost profile. These deployment models can also be mixed to deliver great flexibility when scaling compute and storage resources.

**Supreme Elasticity** – Storage and compute resources can be increased or decreased whenever the need arises. The system automatically rebalances data on the fly. Additions and removals can be done in small or large increments. No capacity planning or complex reconfiguration is required. Unplanned component loss triggers a rebuild operation to preserve data protection. The addition of a component triggers a rebalance to increase available performance and capacity. Rebuild and rebalance operations happen automatically in the background without operator intervention and with no downtime to applications and users.

**Essential Features for Enterprises and Service Providers** – Quality of Service controls permit resource usage to be dynamically managed, limit the amount of performance (IOPS or bandwidth) that selected clients can consume. PowerFlex offers instantaneous, writeable snapshots for data backups and cloning. Operators can create pools with one of two different data layouts to ensure the best environment for workloads. And volumes can be migrated – live and non-disruptively – between different pools should requirements change. Thin provisioning and inline data compression allow for storage savings and efficient capacity management. And with version 3.5, PowerFlex offers native asynchronous replication for Disaster Recovery, data migration, test scenarios, and workload offloading.

PowerFlex provides multi-tenant capabilities via protection domains and storage pools. Protection Domains allow you to isolate specific nodes and data sets. Storage Pools can be used for further data separation, tiering, and performance management. For example, data for performance-demanding business critical applications and databases can be stored in high-performance SSD, NVMe, or SCM-based storage pools for the lowest latency, while less frequently accessed data can be stored in a pool built from low-cost, high-capacity SSDs with lower drive-write-per-day specifications. And, again, volumes can be migrated live from one to another without disrupting your workloads.

# 2 PowerFlex Software Components

PowerFlex fundamentally consists of three types of software components: the Storage Data Server (SDS), the Storage Data Client (SDC), and the Meta Data manager (MDM). Version 3.5 introduces a new component that enables replication, the Storage Data Replicator (SDR).

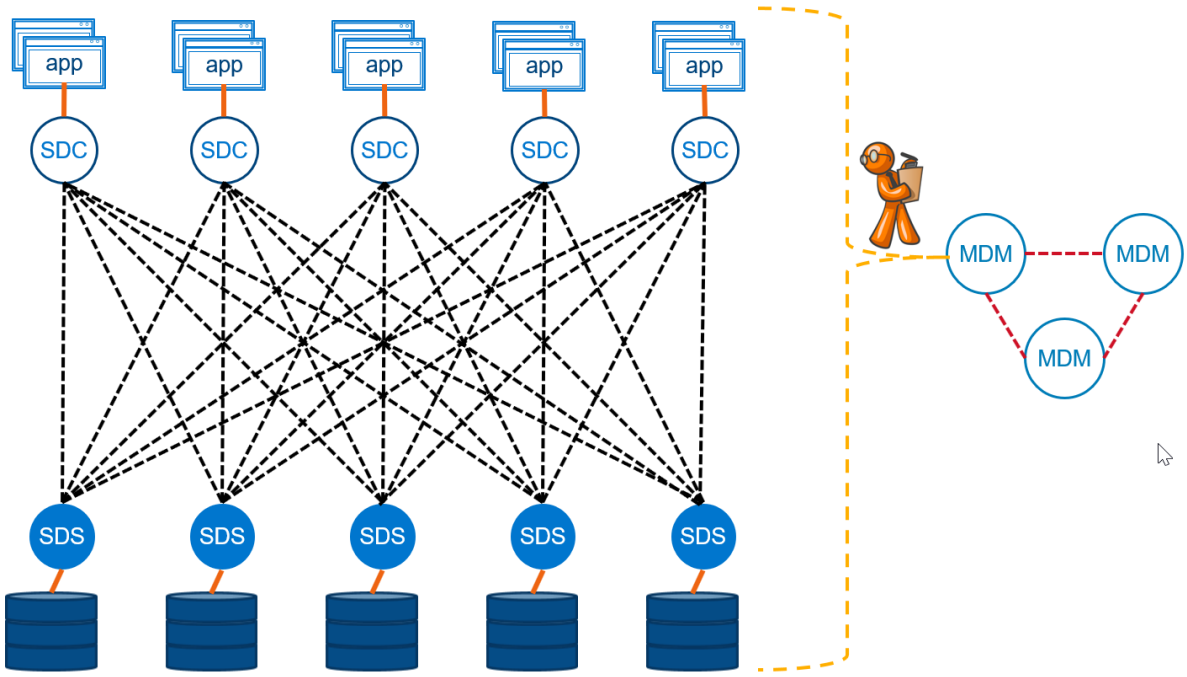


Figure 1 A logical illustration of a PowerFlex deployment. Each volume available to an SDC is distributed across many systems running the SDS, and each SDC has redundant paths to every SDS that services the volume. The Meta Data Manager cluster (MDMs) reside outside the data path where they monitor the system, coordinate data layouts and update SDCs if any changes occur.

## 2.1 Storage Data Server (SDS)

The Storage Data Server (SDS) is a user space service that aggregates raw local storage in a node and serves it out as part of a PowerFlex cluster. The SDS is the server-side software component. Any server that takes part in serving data to other nodes has an SDS service installed and running on it. A collection of SDSs form the PowerFlex persistence layer.

Acting together, SDSs maintain redundant copies of the user data, protect each other from hardware loss, and reconstruct data protection when hardware components fail. SDSs may leverage SSDs, PCIe based flash, Storage Class Memory, spinning disk media, available RAM, or any combination thereof.

SDSs may run natively on various flavors of Linux, or in a virtual appliance on ESXi. A PowerFlex cluster may have up to 512 SDSs.

SDS components can communicate directly with each other, and collections of SDSs are fully meshed. SDSs are optimized for rebuild, rebalance, and I/O parallelism. The user data layout among SDS components is managed through **storage pools, protection domains, and fault sets**.

Client volumes used by the SDCs are placed inside a **storage pool**. Storage pools are used to logically aggregate similar types of storage media at drive-level granularity. Storage pools provide varying levels of storage service distinguished by capacity and performance.

Protection from node, device, and network connectivity failure is managed with node-level granularity through **protection domains**. Protection domains are groups of SDSs in which user data replicas are maintained.

**Fault sets** allow very large systems to tolerate multiple simultaneous node failures by preventing redundant copies from residing in a set of nodes (for example a whole rack) that might be likely to fail together.

## 2.2 Storage Data Client (SDC)

The Storage Data Client (SDC) allows an operating system or hypervisor to access data served by PowerFlex clusters. The SDC is a client-side software component that can run natively on Windows®, various flavors of Linux, IBM AIX®, ESXi® and others. It is analogous to a software HBA, but it is optimized to use multiple network paths and endpoints in parallel.

The SDC provides the operating system or hypervisor running it with access to logical block devices called “volumes”. A volume is analogous to a LUN in a traditional SAN. Each logical block device provides raw storage for a database or a file system and appears to the client node as a local device.

The SDC knows which Storage Data Server (SDS) endpoints to contact based on block locations in a volume. The SDC consumes the distributed storage resources directly from other systems running PowerFlex. SDCs do not share a single protocol target or network end point with other SDCs. SDCs distribute load evenly and autonomously.

The SDC is extremely lightweight. SDC to SDS communication is inherently multi-pathed across all SDS storage servers contributing to the storage pool. This stands in contrast to approaches like iSCSI, where multiple clients target a single protocol endpoint. The widely distributed character of SDC communications enables much better performance and scalability.

The SDC allows shared volume access for uses such as clustering. The SDC does not require an iSCSI initiator, a fiber channel initiator, or an FCoE initiator. The SDC is optimized for simplicity, speed, and efficiency. A PowerFlex cluster may have up to 1024 SDCs.

## 2.3 Meta Data Manager (MDM)

MDMs control the behavior of the PowerFlex system. They determine and publish the mapping between clients and their volume data; they keep track of the state of the system; and they issue rebuild and rebalance directives to SDS components.

MDMs establish the notion of quorum in PowerFlex. They are the only tightly clustered component of PowerFlex. They are authoritative, redundant, and highly available. They are not consulted during I/O operations or during SDS to SDS operations like rebuilding and rebalancing. Although, when a hardware component fails, the MDM cluster will instruct an auto-healing operation to begin within seconds. An MDM cluster is comprised of at least three servers, to maintain quorum, but five can be used to improve availability. In either the 3- or 5-node MDM cluster, there is always one Primary. There may be one or two secondary MDMs and one or two Tie Breakers.

## 2.4 Storage Data Replicator (SDR)

Starting with version 3.5, a new, optional, piece of software is introduced that facilitates asynchronous replication between PowerFlex clusters. The Storage Data Replicator (SDR) is not required for general PowerFlex operation if replication is not employed. On the source side, the SDR stands as a middle-man between an SDC and the SDSs hosting the relevant parts of a volume's address space. When a volume is being replicated, the SDC sends writes to the SDR where the writes are split, and both written to a replication Journal and forwarded to the relevant SDS service for committal to local disk.

SDRs accumulate writes in an interval-journal until the MDM instructs for the interval to be closed. If a volume is a part of a multi-volume Replication Consistency Group, then the interval closures happen simultaneously. Write folding is applied and the interval is added to the transfer queue for transmission to the target side.

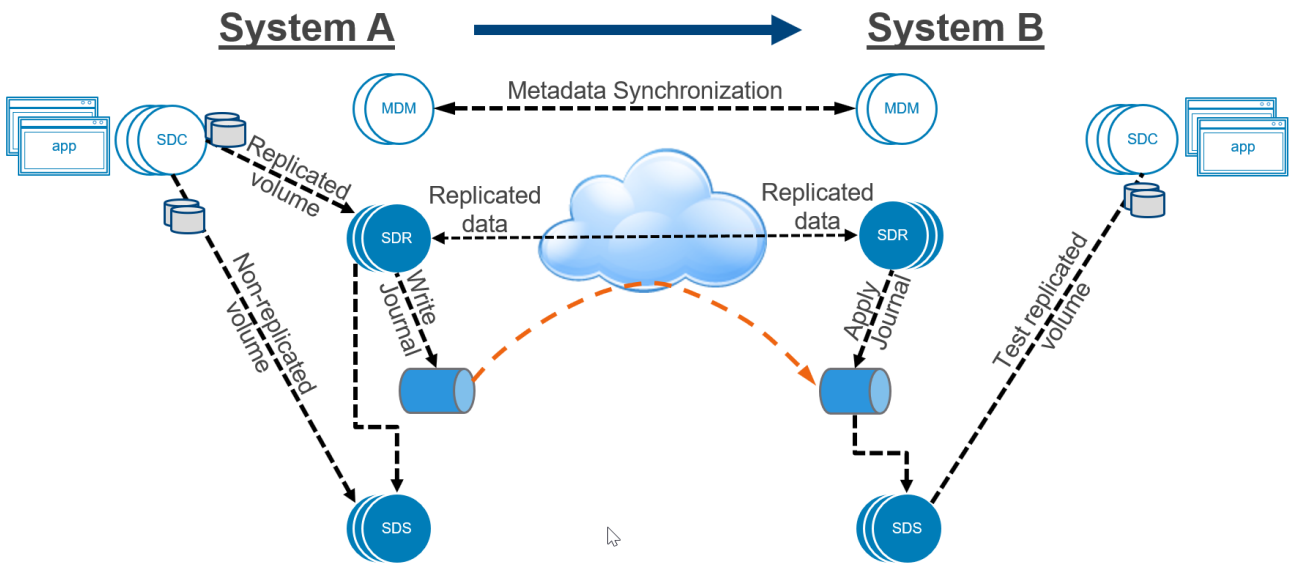


Figure 2 Simplified diagram of replication data flows.

On the target side, the SDR receives the data to another journal and sends it to the SDSs for application to the target replica volume.

# 3 Traffic Types

PowerFlex performance, scalability, and security benefit when the network architecture reflects PowerFlex traffic patterns. This is particularly true in large PowerFlex deployments. The software components that make up PowerFlex (the SDCs, SDSs, MDMs and SDRs) converse with each other in predictable ways. **Architects designing a PowerFlex deployment should be aware of these traffic patterns in order to make informed choices about the network layout.**

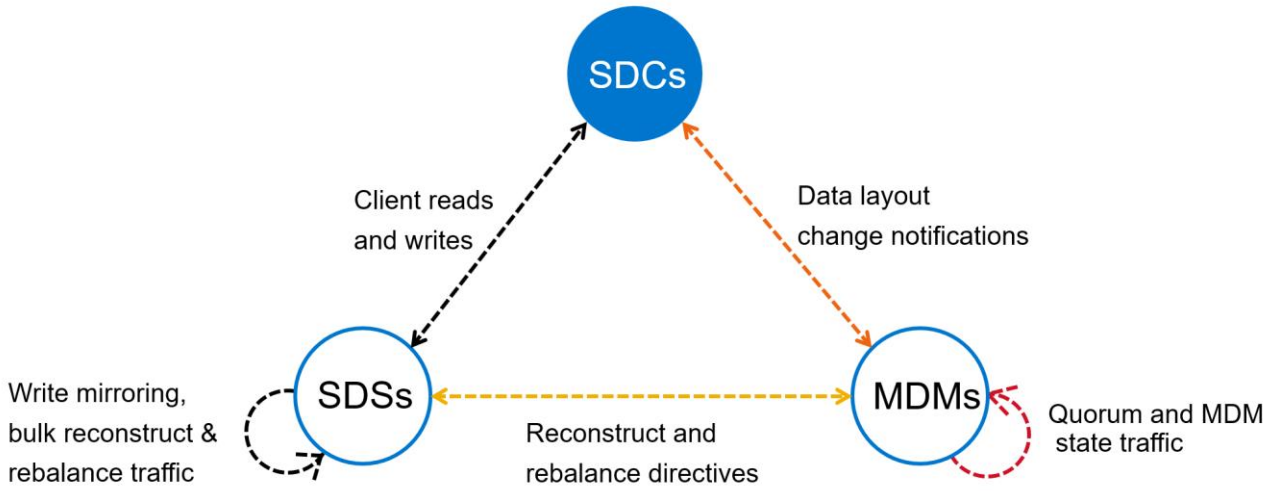


Figure 3 A simplified illustration of how the base PowerFlex software components communicate. A PowerFlex system will have many SDCs, SDSs, and MDMs. This illustration groups SDCs, SDSs, and MDMs. The arrows from the SDSs and MDMs pointing back to themselves represent communication to other SDSs and MDMs. Note that there is no SDC to SDC communication. The traffic patterns are the same regardless of the physical location of an SDC, SDS, or MDM.

In the following discussion, we distinguish front-end traffic from back-end traffic. This is a logical distinction and does not require physically distinct networks. PowerFlex permits running both front-end and back-end traffic over the same physical networks or separating them on to distinct networks. Although not required, isolating front-end and back-end traffic for the storage network is often preferred.

For example, such separation may be done for operational reasons, wherein separate teams manage distinct parts of the infrastructure. The most common reason to separate back-end traffic, however, is that it allows for improved rebuild and rebalance performance. This also isolates front-end traffic, avoiding contention on the network, and lessening latency effects on client or application traffic during rebuild/rebalance operations.

## 3.1 Storage Data Client (SDC) to Storage Data Server (SDS)

Traffic between the SDCs and the SDSs forms the bulk of front-end storage traffic. Front-end storage traffic includes all read and write traffic arriving at or originating from a client. This network has a high throughput requirement.

## 3.2 Storage Data Server (SDS) to Storage Data Server (SDS)

Traffic between SDSs forms the bulk of back-end storage traffic. Back-end storage traffic includes writes that are mirrored between SDSs, rebalance traffic, rebuild traffic, and volume migration traffic. This network has a high throughput requirement.

## 3.3 Meta Data Manager (MDM) to Meta Data Manager (MDM)

MDMs are used to coordinate operations inside the cluster. They issue directives to PowerFlex to rebalance, rebuild, and redirect traffic. They also coordinate Replication Consistency Groups, determine replication journal interval closures, and maintain metadata synchronization with PowerFlex replica-peer systems. MDMs are redundant and must continuously communicate with each other to establish quorum and maintain a shared understanding of data layout.

MDMs do not carry or directly interfere with I/O traffic. The data exchanged among them is relatively lightweight, and MDMs do not require the same level of throughput required for SDS or SDC traffic. However, the MDMs have a very short (<400ms) timeout for their quorum exchanges, which happen every 100ms. **MDM to MDM traffic requires a stable, reliable, low latency network.** MDM to MDM traffic is considered back-end storage traffic. PowerFlex supports the use of one or more networks dedicated to traffic between MDMs. At a minimum, two 10 GbE links should be used per MDM for production environments, although 25GbE is more common.

PowerFlex 3.5 introduces cross-cluster MDM to MDM traffic between replication peer systems. These MDMs must communicate to control replication flow and journal states. They synchronize the consolidated replication states between the source and destination sites. MDM to MDM peer metadata synchronization should take place over a WAN with less than 200ms latency.

## 3.4 Meta Data Manager (MDM) to Storage Data Client (SDC)

The Primary (what the software calls the master) MDM must communicate with SDCs in the event that data layout changes. This can occur because the SDSs that host an SDC's volume(s) storage for the SDCs are added, removed, placed in maintenance mode, or go offline. It may also happen if a volume is placed into a Replication Consistency Group. Communication between the Primary MDM and the SDCs is lazy and asynchronous but still requires a reliable, low latency network. MDM to SDC traffic is considered front-end storage traffic.

## 3.5 Meta Data Manager (MDM) to Storage Data Server (SDS)

The Primary MDM must communicate with SDSs to monitor SDS and device health and to issue rebalance and rebuild directives. MDM to SDS traffic requires a reliable, low latency network. MDM to SDS traffic is considered back-end storage traffic.

## 3.6 Storage Data Client (SDC) to Storage Data Replicator (SDR)

In cases where volumes are replicated, the normal SDC to SDS traffic is routed through the SDR. If a volume is placed into a Replication Consistency Group, the MDM adjusts the volume mapping presented to the SDC and directs the SDC to issue I/O operations to SDRs, which then pass it on to the relevant SDSs. The SDR appears to the SDC as if it were just another SDS. SDC to SDR traffic has a high throughput requirement and requires a reliable, low latency network. SDC to SDR traffic is considered front-end storage traffic.

## 3.7 Storage Data Replicator (SDR) to Storage Data Server (SDS)

When volumes are being replicated and I/O is sent from the SDC to the SDR, there are two subsequent I/Os from the SDR to SDSs on the source system. First the SDR passes on the volume I/O to the associated SDS for processing (e.g., compression) and committal to disk. Second, the SDR applies writes to the journaling volume. Because the journal volume is just another volume in a PowerFlex system, the SDR is sending I/O to the SDSs whose disks comprise the storage pool in which the journal volume resides.

On the target system, the SDR applies the received, consistent journals to the SDSs backing the replica volume. In each of these cases, the SDR behaves as if it were an SDC. Nevertheless, SDR to SDS traffic is considered back-end storage traffic. SDR to SDS traffic throughput may be high and is proportionate to the number of volumes being replicated. It requires a reliable, low latency network.

## 3.8 Metadata Manager (MDM) to Storage Data Replicator (SDR)

MDMs must communicate with SDRs to issue journal-interval closures, collect and report RPO compliance, and maintain consistency at destination volumes. Using the replication state transmitted from peer systems, the MDM commands its local SDRs to perform journal operations.

## 3.9 Storage Data Replicator (SDR) to Storage Data Replicator (SDR)

SDRs within a source or within a target PowerFlex cluster do not communicate with one another. But SDRs in a source system will communicate with SDRs in a replica target system. SDRs ship journal intervals over LAN or WAN networks to destination SDRs. Latency is not as sensitive in SDR → SDR traffic, but round-trip time should not be greater than 200ms**.**

## 3.10 Other Traffic

There are many other types of low-volume traffic in a PowerFlex cluster. Other traffic includes infrequent management, installation, and reporting. This also includes traffic to the PowerFlex Gateway (REST API Gateway, Installation Manager, and SNMP trap sender), the vSphere Plugin, PowerFlex Manager, traffic to and from the Light Installation Agent (LIA), and reporting or management traffic to the MDMs (such as syslog for reporting and LDAP for administrator authentication). It also includes CHAP authentication traffic among the MDMs the SDSs and SDCs. See the “Getting to Know Dell EMC PowerFlex” Guide in the PowerFlex Technical Resource Center for more.

SDCs do not communicate with other SDCs. This can be enforced using private VLANs and network firewalls.

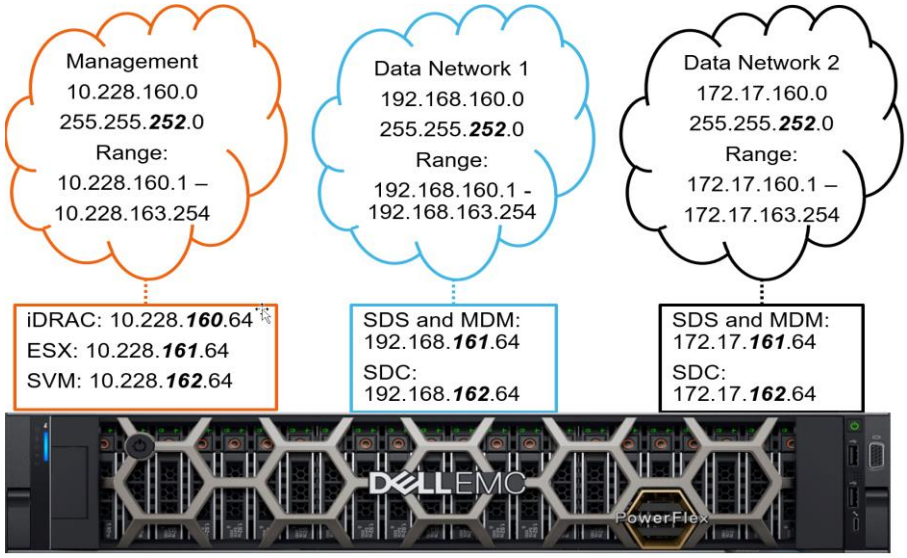


Figure 4 A simple PowerFlex hyperconverged network layout. The management network is routed, and provides access to the iDRAC, ESX, and the Storage Virtual Machine (SVM). Redundant networks carry SDS, MDM, and SDC traffic. The SDS and MDM traffic use the same set of IP addresses. The traffic is not segmented into front-end (SDS, SDC, MDM) and back-end traffic (SDS, MDM), as might be the case in a larger deployment. The 192.168.160.X and 172.17.160.X address spaces can be used for the MDM virtual IP.

# 4 PowerFlex TCP port usage

PowerFlex operates over an Ethernet fabric. While many PowerFlex protocols are proprietary, all communications use standard TCP/IP transport.

The following diagram provides a high-level overview of the port usage and communications among the PowerFlex software components. Some ports are fixed and may not be changed, while others are configurable and may be reassigned to a different port. For a full listing and categorization, see the “Port usage and change default ports” section of the Dell EMC PowerFlex Security Configuration Guide.

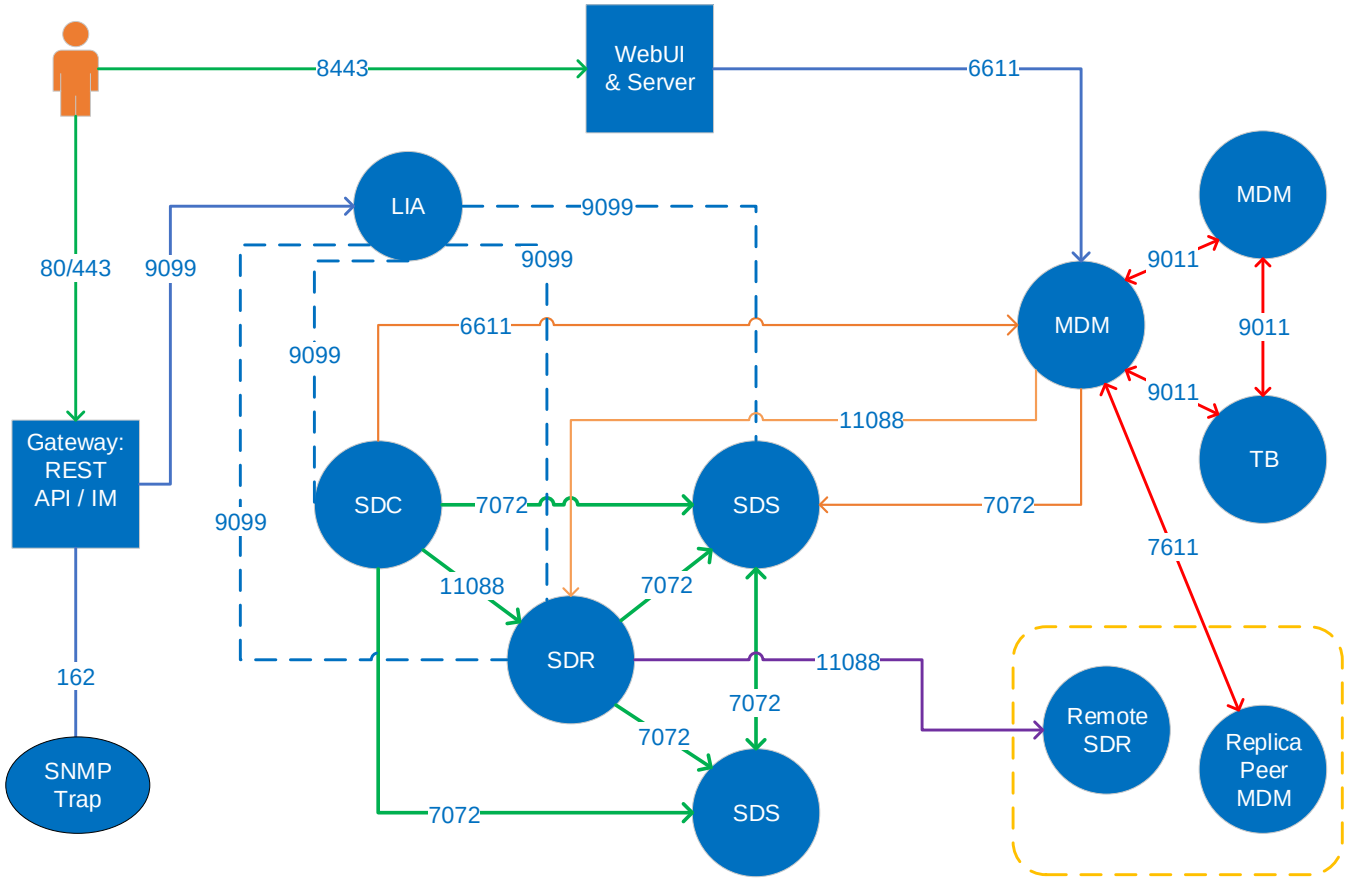


Figure 5 TCP port usage and communications within PowerFlex software-defined storage components. Arrows in the diagram indicate the direction of connection initiation. That is, the arrow points to a listening service port. Data may travel both directions over a connection after initiation. Dashed lines indicate that communication is internal to a node, among installed components.

Ports 25620 and 25600 on the MDM and 25640 on the SDS may also be listening. These are used only by PowerFlex internal debugging tools and are not a part of daily operation and traffic.

# 5 Network Fault Tolerance

Communications between PowerFlex components (MDM, SDS, SDC, SDR) should be assigned to at least two subnets on different physical networks. The PowerFlex networking layer of each of these components provides native link fault tolerance and multipathing across the multiple subnets assigned. There are advantages by-design resulting from this:

1. In the event of a link failure, PowerFlex becomes aware of the problem almost immediately, and adjusts to the loss of bandwidth.
2. If switch-based link aggregation were used, PowerFlex has no means of identifying a single link loss.
3. PowerFlex will dynamically adjust communications within 2–3 seconds across the subnets assigned to the MDM, SDS, and SDC components when a link fails. This is particularly important for SDS→SDS and SDC→SDS connections.
4. Each of these components has the ability to load balance and aggregate traffic across up to eight subnets, reducing the complexity of maintaining switch-based link aggregation. And, because it is managed by the storage layer itself, can be more efficient and simpler to maintain than switch-based aggregation.

   **Note:** In previous versions of PowerFlex software, if a link related failure occurred, there could be a network service interruption and I/O delay of up to 17 seconds in the SDC→SDS networks. The SDC has a general 15-second timeout, and I/O would only be reissued on another “good” socket when the timeout had been reached and the dead socket is already closed.

   In version 3.5 and forward, PowerFlex no longer relies upon I/O timeouts but uses the link disconnection notification. After a link down event, all the related TCP connections are closed after 2 seconds, and all in-flight I/O messages that have not received a response are aborted and the I/Os are reissued by the SDC.

Both native network path load balancing and switch-based link aggregation are fully supported, but it is often simpler to rely on native network path load balancing. If desired, the approaches can be combined to create, for example, two data-path networks over a trunk where each logical network has use of two physical ports per node.

PowerFlex Manager does exactly this for the appliance. It uses link aggregation in combination with the native multipathing to provide layered and robust network fault tolerance. See the Dell EMC PowerFlex Appliance Network Planning Guide.

# 6 Network Infrastructure

Leaf-spine and flat network topologies are the most commonly used with PowerFlex today. Flat networks are used in smaller networks. In modern datacenters, leaf-spine topologies are preferred over legacy hierarchical topologies. This section compares flat and leaf-spine topologies as a transport medium for PowerFlex data traffic.

**Dell Technologies recommends the use of a non-blocking network design.** Non-blocking network designs allow the use of all switch ports concurrently, without blocking some of the network ports to prevent message loops. Therefore, Dell Technologies strongly recommends against the use of Spanning Tree Protocol (STP) on a network hosting PowerFlex. In order to achieve maximum performance and predictable quality of service, the network should not be over-subscribed.

## 6.1 Leaf-Spine Network Topologies

A two-tier leaf-spine topology provides a single switch hop between leaf switches and provides a large amount of bandwidth between end points. A properly sized leaf-spine topology eliminates oversubscription of uplink ports. Very large datacenters may use a three-tier leaf-spine topology. For simplicity, this paper focuses on two tier leaf-spine deployments.

In a leaf-spine topology, each leaf switch is attached to all spine switches. Leaf switches do not need to be directly connected to other leaf switches. Spine switches do not need to be directly connected to other spine switches.

In most instances, Dell Technologies recommends using a leaf-spine network topology. This is because:

- PowerFlex can scale out to many hundreds of nodes in a single cluster.
- Leaf-spine architectures are future proof. They facilitate scale-out deployments without having to re-architect the network.
- A leaf-spine topology allows the use of all network links concurrently. Legacy hierarchical topologies must employ technologies like Spanning Tree Protocol (STP), which blocks some ports to prevent loops.
- Properly sized leaf-spine topologies provide more predictable latency due to the elimination of uplink oversubscription.

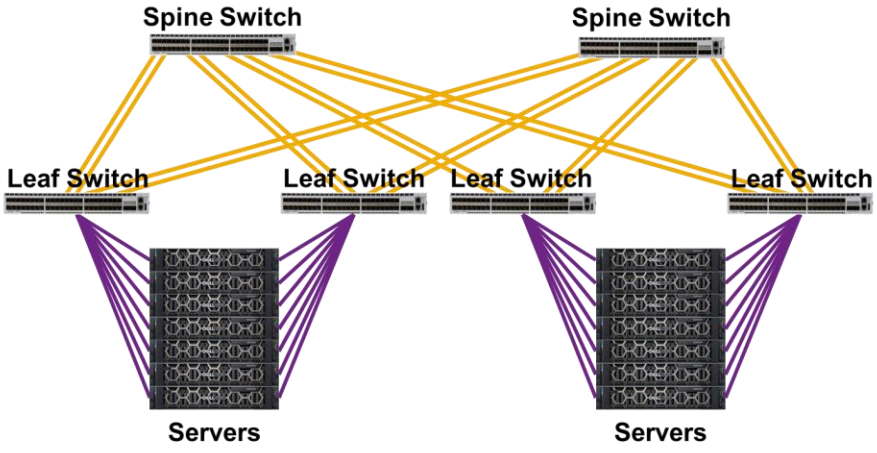


Figure 6 A two-tier leaf-spine network topology. Each leaf switch has multiple paths to every other leaf switch. All links are active. This provides increased throughput between devices on the network. Leaf switches may be connected to each other for use with MLAG (not shown).

## 6.2 Flat Network Topologies

A flat network topology can be easier to implement and may be the preferred choice if an existing flat network is being extended or if the network is not expected to scale. In a flat network, all the switches are used to connect hosts. There are no spine switches.

If you expand beyond a small number of access switches, however, the additional cross-link ports required could likely make a flat network topology cost prohibitive. Use-cases for a flat network topology include Proof-of-Concept deployments and small datacenter deployments that will not grow beyond a few racks.

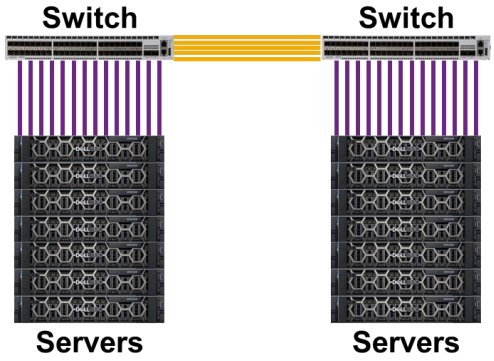


Figure 7 A flat network. This network design reduces cost and complexity at the expense of redundancy and scalability. In this visualization, each switch is a single point of failure. It is possible to build a flat network without a single point of failure using technology such as MLAG (not shown).

# 7 Network Performance and Sizing

A properly sized network frees network and storage administrators from concerns over individual ports or links becoming performance or operational bottlenecks. The management of networks instead of endpoint hot-spots is a key architectural advantage of PowerFlex.

Because PowerFlex distributes I/O evenly across multiple points in a network, network performance must be sized appropriately.

## 7.1 Network Latency

Network latency is important to account for when designing your network. Minimizing the amount of network latency will provide for improved performance and reliability. **For best performance, latency for all SDS and SDC communication should never exceed 1 millisecond network-only round-trip time under normal operating conditions.** Since a wide-area network's (WAN's) lowest response times generally exceed this limit, you should not operate PowerFlex clusters across a WAN.

Systems implementing asynchronous replication are not an exception to this with respect to general, SDC, MDM and SDS communications. Data is replicated between independent PowerFlex clusters, each of which should itself adhere to the sub-1ms rule. The difference is the latency between the peered systems. Because asynchronous replication usually takes place over WAN, the latency requirements are necessarily less restrictive. **Network latency between peered PowerFlex cluster components, however, whether MDM←→MDM or SDR←→SDR, should not exceed 200ms round trip time.**

Latency should be tested in both directions between all components. This can be verified by pinging, and more extensively by the SDS Network Latency Meter Test. The open source tool iPerf can be used to verify bandwidth. Please note that iPerf is not supported by Dell Technologies. iPerf and other tools used for validating a PowerFlex deployment are covered in detail in the "Validation Methods" section of this document.

## 7.2 Network Throughput

Network throughput is a critical component when designing your PowerFlex implementation. Throughput is important to reduce the amount of time it takes for a failed node to rebuild; to reduce the amount of time it takes to redistribute data in the event of uneven data distribution; to optimize the amount of I/O a node is capable of delivering; and to meet performance expectations.

While PowerFlex software can be deployed on a 1-gigabit network for test or investigation purposes, storage performance will likely be bottlenecked by network capacity. **At a bare minimum, Dell recommends leveraging 10-gigabit network technology, with 25-gigabit technology as the preferred minimum link throughput.** All current PowerFlex nodes ship with at least four ports, each at a minimum port bandwidth of 25GbE, with 100GbE ports offered as the forward-looking option. This is especially important when considering replication cases and their additional bandwidth requirements.

Additionally, although the PowerFlex cluster itself may be heterogeneous, the **SDS components that make up a protection domain should reside on hardware with equivalent storage and network performance**. This is because the total bandwidth of the protection domain will be limited by the weakest link during I/O and

reconstruct/rebalance operations due to the wide striping of volume data across all contributing components. Think of it like a hiking party able to travel no faster than its slowest member.

A similar consideration holds when mixing heterogeneous OS and hypervisor combinations. VMware-based hyperconverged infrastructure has a slower performance profile than bare-metal configurations due to the virtualization overhead, and mixing HCI and bare metal nodes in a protection domain will limit the throughput of storage pools containing both to the performance capability of the slowest member. It is possible and allowed (from the storage software perspective), but the user must take note of this implication. It is not a supported configuration for the PowerFlex rack or appliance.

In addition to throughput considerations, **it is recommended that each node have at least two separate network connections for redundancy, regardless of throughput requirements.** This remains important even as network technology improves. For instance, replacing two 40-gigabit links with a single 100-gigabit link improves throughput but sacrifices link-level network redundancy.

In most cases, the amount of network throughput to a node should match or exceed the combined maximum throughput of the storage media hosted on the node. *Stated differently, a node's network requirements are proportional to the total performance of its underlying storage media.*

When determining the amount of network throughput required, keep in mind that modern media performance is typically measured in *megabytes* per second, but modern network links are typically measured in *gigabits* per second.

To translate *megabytes* per second to *gigabits* per second, first multiply *megabytes* by 8 to translate to *megabits*, and then divide *megabits* by 1,000 to find *gigabits*.

$$\text{gigabits} = \frac{\text{megabytes} * 8}{1{,}000}$$

Note that this is not perfectly precise, as it does not account for the base-2 definition of "kilo" as 1024, which is standard in PowerFlex, but it is adequate for this paper's explanatory purposes.

### 7.2.1 Example: An SDS-only (storage only) node with 10 SSDs

Assume that you have a 1U node hosting only an SDS. This is not a hyper-converged environment, so only storage traffic must be considered. The node contains 10 SAS SSD drives. Each of these drives is individually capable of delivering a raw throughput of 1000 megabytes per second under the best conditions (sequential I/O, which PowerFlex is optimized for during reconstruct and rebalance operations). The total throughout of the underlying storage media is therefore 10,000 *megabytes* per second.

$$10 * 1000 \text{ megabytes} = 10{,}000 \text{ megabytes}$$

Then convert 10,000 *megabytes* to *gigabits* using the equation described earlier: first multiply 10,000MB by 8, and then divide by 1,000.

$$\frac{10{,}000 \text{ megabytes} * 8}{1{,}000} = 80 \text{ gigabits}$$

In this case, if all the drives on the node are serving read operations at the maximum speed possible, the total network throughput required would be 80 gigabits per second. We are accounting for read operations only, which is typically enough to estimate the network bandwidth requirement. This cannot be serviced by a single

25- or 40-gigabit link, although theoretically a 100GbE link would suffice. However, since network redundancy is encouraged, this node should have at least two 40 gigabit links, with the standard 4x 25GbE configuration preferred.

**Note:** calculating throughput based only on the theoretical throughput of the component drives may result in unreasonably high estimates for a single node. **Verify that the RAID controller or HBA on the node can also meet or exceed the maximum throughput of the underlying storage media.**

### 7.2.2 Write-heavy environments

Read and write operations produce different traffic patterns in a PowerFlex environment. When a host (SDC) makes a single 4k read request, it must contact a single SDS to retrieve the data. The 4k block is transmitted once, out of a single SDS. If that host makes a single 4k write request, the 4k block must be transmitted to the primary SDS, then copied out of the primary SDS to the secondary SDS.

Write operations therefore require two times more bandwidth to SDSs than read operations. However, a write operation involves two SDSs, rather than the one required for a read operation. The bandwidth requirement ratio of reads to writes is therefore 1:1.5.

Stated differently, per SDS, a write operation requires 1.5 times more network throughput than a read operation when compared to the throughput of the underlying storage.

Under ordinary circumstances, the storage bandwidth calculations described earlier are sufficient. **However, if some of the SDSs in the environment are expected to host a write-heavy workload, consider adding network capacity.**

### 7.2.3 Environments with volumes replicated to another system

Version 3.5 introduces native asynchronous replication, which must be accounted for when considering the bandwidth generated, first, within the cluster and, second, between replica peer systems.

#### 7.2.3.1 Bandwidth within a replicating system

We noted above that when a volume is being replicated I/O is sent from the SDC to the SDR, after which there are subsequent I/Os from the SDR to SDSs on the source system. The SDR first passes on the volume I/O to the associated SDS for processing (e.g., compression) and committal to disk. The associated SDS will probably not be on the same node as the SDR, and bandwidth calculations must account for this. In the second step, the SDR applies incoming writes to the journaling volume. Because the journal volume is just like any other volume within a PowerFlex system, the SDR is sending I/O to the various SDSs backing the storage pool in which the journal volume resides. *This step adds two additional I/Os as the SDR first writes to the relevant primary SDS backing the journal volume and the primary SDS sends a copy to the secondary SDS.* Finally, the SDR makes an extra read from the journal volume before sending to the remote site.

Write operations for replicated volumes therefore require three times as much bandwidth within the source cluster as write operations for non-replicated volumes. **Carefully consider the write profile of workloads that will run on replicated volumes; additional network capacity will be needed to accommodate the additional write overhead**. In replicating systems, therefore, we recommend using 4x 25GbE or 2x 100GbE networks to accommodate the back-end storage traffic.

#### 7.2.3.2 Bandwidth between replica peer systems

Turning to consider network requirements between replica peer systems, we reiterate that **there should be no more than 200ms latency between source and target systems.**

Journal data is shipped between source and target SDRs, first, at the replication pair initialization phase and, second, during the replication steady state phase. Special care should be taken to ensure adequate bandwidth between the source and target SDRs, whether over LAN or WAN. The potential for exceeding available bandwidth is greatest over WAN connections. While write-folding may reduce the amount of data to be shipped to the target journal, this cannot always be easily predicted. *If the available bandwidth is exceeded, the journal intervals will back up, increasing both the journal volume size and the RPO.*

**As a best practice, we recommend that the sustained write bandwidth of all volumes being replicated should not exceed 80% of the total available WAN bandwidth.** if the peer systems are mutually replicating volumes to one another, the peer SDR←→SDR bandwidth must account for the requirements of both directions simultaneously. Reference and use the latest PowerFlex Sizer for additional help calculating the required WAN bandwidth for specific workloads.

**Note:** The sizer tool is an internal tool available for Dell employees and partners. External users should consult with their technical sales specialist if WAN bandwidth sizing assistance is needed.

#### 7.2.3.3 Networking implications for replication health

While this paper's focus is PowerFlex networking information best practices, the general operation, health and performance of the storage layer itself depends on the quality and capacity of the networks deployed. This has particular relevance for asynchronous replication and the sizing of journal volumes.

It is possible to have write peaks that exceed the recommended "0.8 * WAN bandwidth", but they should be short. The journal size must be large enough to absorb these write peaks.

This is important. The journal volume capacity should be sized to accommodate link outages between peer systems. A one-hour outage might be reasonably expected, but we strongly encourage users to plan for 3 hours. One must ensure sufficient journal space to account for the application writes during the outage. I**n general, the journal capacity should be calculated as Peak Write Bandwidth * link down time**. We need to know the maximum application write bandwidth during the busiest hour. Let's say our application has a peak write throughput of 1GB/s. 3 hours is 10800 seconds. So, the required journal capacity is

$$1GB/s * 10800 \text{ seconds} = \sim 10.55TB$$

However, PowerFlex sets journal capacity as a percentage of pool capacity. Assuming we have one 200TB storage pool:

$$100 * 10.55TB \ / \ 200TB = 5.27\%$$

As a safety margin, round this up to 6%.

**Note:** The volume data shipped in the journal intervals is not compressed. In PowerFlex, compression is for data at rest. In fine-granularity storage pools, data compression takes place in the SDS service after it has been received from an SDC (for non-replicated volumes) or an SDR (for replicated volumes). The SDR is unaware of and agnostic to the data layout on either side of a replica pair. If the destination, or target, volume is configured as compressed, the compression takes place in the target system SDSs as the journal intervals are being applied.

### 7.2.4 Hyper-converged environments

When PowerFlex is in a hyper-converged deployment, each physical node is running an SDS, an SDC on the hypervisor, and one or more VMs. In this sense, a hyper-converged PowerFlex deployment need not involve a hypervisor. Hyper-converged deployments optimize hardware investments, but they also introduce network sizing requirements.

**The storage bandwidth calculations described earlier apply to hyper-converged environments, but front-end bandwidth to any virtual machines, hypervisor or OS traffic, and traffic from the SDC, must also be considered.** Though sizing for the virtual machines is outside the scope of this technical report, it is a priority.

In hyper-converged environments, it is also a priority to logically separate storage from other network traffic.

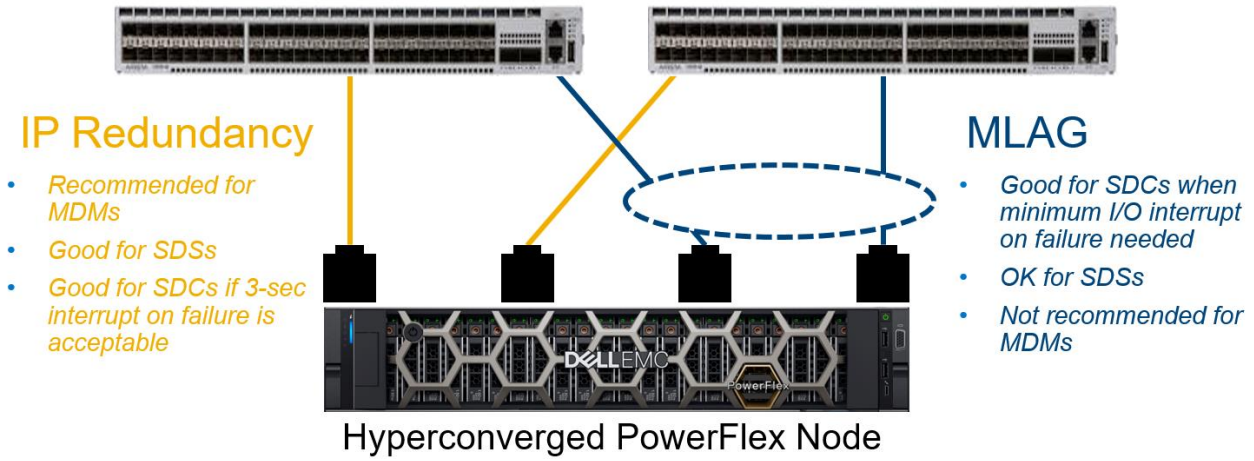


Figure 8 An example of a hyper-converged VMware environment using 4 x 25 gigabit network connections. PowerFlex traffic on this host utilize ports Eth0 and Eth1. Redundancy is provided with native PowerFlex IP multipathing, rather than MLAG. Ports Eth2 and Eth3 use both MLAG and VLAN tagging and provide access network access to the hypervisor and the other guests. Other configurations are possible as PowerFlex supports VLAN tagging and link aggregation.

# 8 Network Hardware

## 8.1 Dedicated NICs

**PowerFlex engineering recommends the use of dedicated network adapters for PowerFlex traffic, if possible**. Dedicated network adapters provide dedicated bandwidth and simplified troubleshooting. Note that shared network adapters are supported and may be mandatory in hyper-converged environments.

## 8.2 Shared NICs

While not optimal, the use of shared NICs is supported by PowerFlex software. If PowerFlex traffic will share physical networks with other non-PowerFlex traffic, QoS should be implemented to avoid network congestion or starvation issues arising from either PowerFlex or the non-PowerFlex traffic.

## 8.3 Two NICs vs. Four NICs and Other Configurations

PowerFlex allows for the scaling of network resources through the addition of additional network interfaces. **Although not required, there may be situations where isolating front-end and back-end traffic for the storage network may be ideal.** This may be useful in two-layer deployments where the storage and virtualization or compute teams each manage their own networks. More commonly, a user will segment front-end and back-end network traffic to guarantee the performance of storage- and application-related network traffic. In all cases, Dell recommends multiple interfaces for redundancy, capacity, and speed.

PCI NIC redundancy is also a consideration. **The use of two dual-port PCI NICs on each server is preferable to the use of a single quad-port PCI NIC, as a two dual-port PCI NICs can be configured to survive the failure of a single NIC.**

## 8.4 Switch Redundancy

In most leaf-spine configurations, spine switches and top-of-rack (ToR) leaf switches are redundant. This provides continued access to components inside the rack in the network in the event a ToR switch fails. In cases where each rack contains a single ToR switch, ToR switch failure will result in an inability to access the SDS components inside the rack. **Therefore, single ToR switch configurations are not recommended.** If a single ToR switch is used per rack, users should define fault sets at the rack level to ensure data availability in the case of switch failure.

# 9 IP Considerations

## 9.1 IPv4 and IPv6

Starting with version 2.6, and included in all versions after 3.0, PowerFlex provides IPv6 support in both the two-layer and hyperconverged deployment options. Earlier versions of PowerFlex supported Internet Protocol version 4 (IPv4) addressing only. The examples in this paper, focus on IPv4.

## 9.2 IP-level Redundancy

MDMs, SDSs, SDRs and SDCs can have multiple IP addresses, and can therefore reside in more than one network. This provides options for load balancing and redundancy.

PowerFlex natively provides redundancy and load balancing across physical network links when a software component is configured to send traffic across multiple links. In this configuration, each physical network port available to the MDM, SDR or SDS is assigned its own IP address, each in a different subnet.

The use of multiple subnets provides redundancy at the network level. The use of multiple subnets also ensures that as traffic is sent from one component to another, a different entry in the source component's route table is chosen depending on the destination IP address. This prevents a single physical network port at the source from being a bottleneck as the source contacts multiple IP addresses (each corresponding to a physical network port) on a single destination.

Stated differently, a bottleneck at the source port may happen if multiple physical ports on the source and destination are in the same subnet. For example, if two SDSs share a single subnet, each SDS has two physical ports, and each physical port has its own IP address in that subnet, the IP stack will cause the source SDS to always choose the same physical source port. **Splitting ports across subnets allows for load balancing, because each port corresponds to a different subnet in the host's routing table.**

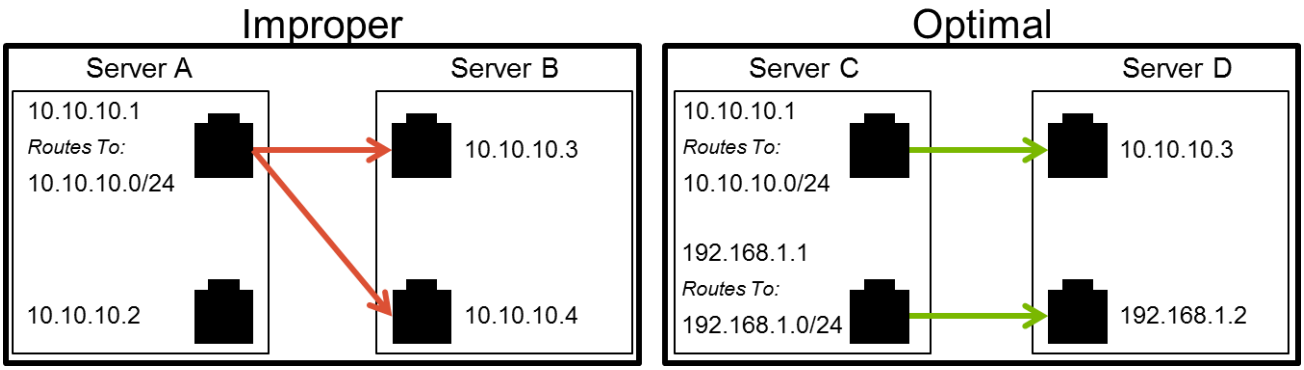


Figure 9 A comparison of operating system IP configurations. The improper IP configuration on the left uses the same subnet, 10.10.10.0/24, for all traffic. When Server A initiates a connection to Server B, the network link providing a route to 10.10.10.0/24 will always be chosen for the outgoing connection. The second network port on Server A will not be utilized for outgoing connections. The proper IP configuration on the right uses two subnets, 10.10.10.0/24 and 192.168.1.0/24, allowing both ports on Server C to be utilized for outgoing connections. Note: the subnets chosen in this example (10.10.10.0/24 and 192.168.1.0/24) are arbitrary: the mixed use of a class "A" and a class "C" is meant for visual distinction only.

When each MDM or SDS has access to multiple IP addresses, PowerFlex will handle load balancing more effectively due to its awareness of the traffic pattern. This can result in a small performance boost. Additionally, link aggregation maintains its own set of timers for link-level failover. Native PowerFlex IP-level redundancy can therefore ease troubleshooting when a link goes down.

IP-level redundancy also protects against IP address conflicts. To protect against unwanted IP changes or conflicts, **DHCP must not be deployed on a network where PowerFlex MDMs or SDCs reside.**

**When used in isolation, IP-level redundancy is strongly preferred over MLAG for links in use for MDM to MDM communication.** If IP-level redundancy is layered in VLANs on top of redundant Link Aggregation Groups, that is a good use of both technologies. See the Dell EMC PowerFlex Appliance Network Planning Guide for examples of this.

# 10 Ethernet Considerations

## 10.1 Jumbo Frames

PowerFlex supports jumbo frames, and using jumbo frames for the storage traffic is highly encouraged. However, enabling jumbo frames can be challenging depending on your network infrastructure. Inconsistent implementation of jumbo frames by the various network components can lead to performance problems that are difficult to troubleshoot. When jumbo frames are in use, they must be enabled on every network component used by PowerFlex infrastructure, including the hosts and switches, and storage VMs if HCI is deployed.

Enabling jumbo frames allows more data to be passed in a single Ethernet frame. This decreases the total number of Ethernet frames and the number of interrupts that must be processed by each node. If jumbo frames are enabled on every component in your PowerFlex infrastructure, there is a performance benefit of approximately 10%, depending on your workload.

**Note:** When PowerFlex Manager is used to deploy a PowerFlex cluster on an appliance or rack system, configuration of jumbo frames on the node and switch components is fully coordinated and managed for all cluster components.

Carefully review the network components to ensure consistent configuration of jumbo frames at every point. If you are uncertain, we recommend leaving jumbo frames disabled initially. Enable jumbo frames only after you have a stable working setup and confirmed that your infrastructure can support their use. To ensure that jumbo frames are configured on all nodes along each path, you can employ utilities like the Linux `tracepath` command to discover MTU sizes along a path. Ping can be useful in diagnosing Jumbo Frame issues as well. On Linux, use the command of the form: `ping -M do -s 8972 <ip address/hostname>`. (Note that here we are subtracting 28 bytes for un-encapsulated packet headers from the 9000 MTU size.)

Refer to the PowerFlex Configure and Customize guide for additional information about implementing jumbo frames.

## 10.2 VLAN Tagging

PowerFlex is agnostic to native VLANs and VLAN tagging on the connection between the server and the access or leaf switch. Being configured in the operating system or switch, these are transparent to PowerFlex software. When measured by PowerFlex engineering, VLANs have no impact on the level of performance.

For the PowerFlex appliance deployment, we expect a standard set of uniform VLANs are configured. See section 19 below.

# 11 Link Aggregation Groups

Link Aggregation Groups (LAGs) and Multi-Chassis Link Aggregation Groups (MLAGs) combine ports between end points. The end points can be a switch and a host with LAG or two switches and a host with MLAG. Link aggregation terminology and implementation varies by switch vendor. MLAG functionality on Cisco Nexus switches is called Virtual Port Channels (vPC).

LAGs use the Link Aggregation Control Protocol (LACP) for setup, tear down, and error handling. LACP is a standard, but there are many proprietary variants.

Regardless of the switch vendor or the operating system hosting PowerFlex, **LACP is recommended when link aggregation groups are used. The use of static link aggregation is not supported.**

Link aggregation can be used as an alternative to IP-level redundancy, where each physical port has its own IP address. Link aggregation can be simpler to configure for some teams and is useful in situations where IP address exhaustion is an issue. Link aggregation must be configured on both the node running PowerFlex and the network equipment it is attached to.

PowerFlex is resilient and high performance regardless of the choice of IP-level redundancy or link aggregation. Performance of SDSs when MLAG is in use is close to the performance of IP-level redundancy.

- **The choice of MLAG or IP-level redundancy for SDSs should be considered an operational decision.**
- **With MDM to MDM traffic, IP-level redundancy or LAG is strongly recommended over MLAG, as the continued availability of one IP address on the MDM helps prevent failovers, due to the short timeouts between MDMs, which are designed to communicate between multiple IP addresses.**
- **Due to improved network failure resiliency in 3.5, IP-level redundancy is generally preferred over MLAG for links in use by SDC components.**

## 11.1 LACP

LACP sends a message across each physical network link in the aggregated group of network links on a periodic basis. This message is part of the logic that determines if each physical link is still active. The frequency of these messages can be controlled by the network administrator using LACP timers.

LACP timers can typically be configured to detect link failures at a fast rate (one message per second) or a normal rate (one message every 30 seconds). When an LACP timer is configured to operate at a fast rate, corrective action is taken quickly. Additionally, the relative overhead of sending a message every second is small with modern network technology.

**LACP timers should be configured to operate at a fast rate when link aggregation is used between a PowerFlex SDS and a switch.**

To establish an LACP connection, one or both of the LACP peers must be configured to use active mode. **It is therefore recommended that the switch connected to the PowerFlex node be configured to use active mode across the link.**

## 11.2 Load Balancing

When multiple network links are active in a link aggregation group, the endpoints must choose how to distribute traffic between the links. Network administrators control this behavior by configuring a load balancing method on the end points. Load balancing methods typically choose which network link to use based on some combination of the source or destination IP address, MAC address, or TCP/UDP port.

This load-balancing method is referred to as a “hash mode”. Hash mode load balancing aims to keep traffic to and from a certain pair of source and destination addresses or transport ports on the same physical link, provided that link remains active.

The recommended configuration of hash mode load balancing depends on the operating system in use.

**If a node running an SDS has aggregated links to the switch and is running VMware ESX®, the hash mode should be configured to use “Source and destination IP address” or “Source and destination IP address and TCP/UDP port”.**

**If a node running an SDS has aggregated links to the switch and is running Linux, the hash mode on Linux should be configured to use the "xmit_hash_policy=layer2+3" or "xmit_hash_policy=layer3+4" bonding option.** The "xmit_hash_policy=layer2+3" bonding option uses the source and destination MAC and IP addresses for load balancing. The "xmit_hash_policy=layer3+4" bonding option uses the source and destination IP addresses and TCP/UDP ports for load balancing.

**On Linux, the “miimon=100” bonding option should also be used.** This option directs Linux to verify the status of each physical link every 100 milliseconds.

Note that the name of each bonding option may vary depending on the Linux distribution, but the recommendations remain the same.

## 11.3 Multiple Chassis Link Aggregation Groups

Like link aggregation groups (LAGs), MLAGs provide network link redundancy. Unlike LAGs, MLAGs allow a single end point (such as a node running PowerFlex) to be connected to multiple switches. Switch vendors use different names when referring to MLAG, and MLAG implementations are typically proprietary.

The use of MLAG is supported by PowerFlex but is not generally recommended for MDM to MDM traffic. See, however, the notes in the following section. The options described in the “Load Balancing” section also apply to the use of MLAG.

# 12 The MDM Network

Although MDMs do not reside in the data path between hosts (SDCs) and their distributed storage (SDSs), they are responsible for maintaining relationships between themselves to keep track of the state of the cluster. MDM to MDM traffic is therefore sensitive to network events that impact latency, such as the loss of a physical network link in an MLAG.

MDMs are redundant. PowerFlex can therefore survive not just an increase in latency, but loss of MDMs. The use of MLAG to a node hosting an MDM will work. **However, if you require the use of MLAG on a network that carries MDM to MDM traffic, please work with a Dell EMC PowerFlex representative to ensure you have chosen a robust design that employs double network redundancy, combining MLAG with native IP-level redundancy.**

**In most situations, it is recommended that MDMs use IP-level redundancy on two or more network segments rather than MLAG.** The MDMs may share one or more dedicated MDM cluster networks.

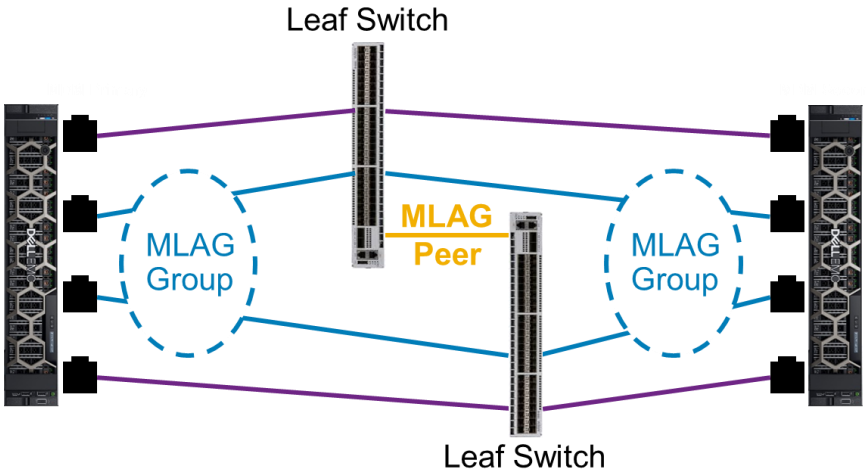


Figure 10 Two nodes connected to two leaf switches. MDM traffic should traverse the purple links because they are not in an MLAG group.

# 13 Network Services

## 13.1 DNS

The MDM cluster maintains the database of system components and their IP addresses. In order to eliminate the possibility of a DNS outage impacting a PowerFlex deployment, the MDM cluster does not track system components by hostname or fully qualified domain name (FQDN). If a hostname or FQDN is used when registering a system component with the MDM cluster, it is resolved to an IP address and the component is registered with its IP address.

The exception to this is when the VASA provider is deployed and vVols are implemented. The use of vVols in a PowerFlex environment requires the deployment of the PowerFlex VASA provider (in either single mode or a 3-node cluster). Implementing vVols technology into a vSphere environment requires fully FQDNs for the vCenter server, the ESXi hosts which will use vVol datastores, and the VASA provider hosts themselves. There must be valid DNS resolution among all of these components. The DNS service employed must therefore be highly available to prevent loss of vVol connectivity and functionality.

In summary, **hostname and FQDN changes do not generally influence inter-component traffic in a PowerFlex deployment unless vVols are implemented.**

# 14 Replication Network over WAN

There are additional considerations to account for when using PowerFlex native asynchronous replication. In sections 2.4 and 3.9, we covered the Storage Data Replicator (SDR) and its traffic. In section 7.2.3, we covered additional bandwidth requirements. In this section, we consider addressing and routing topics specific to running replication over a wide area network (WAN). The recommendations are general, as implementation details depend on the hardware and WAN topology used.

## 14.1 Additional IP addresses

Within a protection domain, SDRs are installed on the same hosts as SDSs, but the traffic that an SDR writes to a journal volume is sent to all SDSs that host the journal, not only the one is it co-located with on a host. In the backend storage network, each SDR listens on the same node IPs as the SDSs and therefore should be able to reach all SDSs in the protection domain.

The SDRs, however, require additional, distinct IP addresses which will allow them to communicate with remote SDRs. In most cases, these should be routable addresses with a properly configured gateway. For redundancy, each SDR should have two.

## 14.2 Firewall Considerations

SDRs communicate with each other, and ship replicated data between themselves, over TCP port 1088. This port must be open for egress in any firewall on the source system side, and it must be open for ingress on the target system side. If replication is being performed in both directions between two systems, then port 1088 must be open in the firewall for both egress and ingress on both sides.

## 14.3 Static Routes

PowerFlex asynchronous replication usually happens over a WAN between physically remote clusters that do not share the same address segments. If the default route itself is not suitable to properly direct packets to the remote SDR IPs, static routes should be configured to indicate either the next hop address or the egress interface or both for reaching the remote subnet.

For example: `X.X.X.X/X via X.X.X.X dev interface`

Consider a small system with a few nodes on each side. Each node has four network adapters, two of which are configured with IPs for communication internal to the PowerFlex cluster and two of which are configured with IP addresses for site-to-site, external communication.

In this example, we tell the nodes to access the WAN subnets for the other side through a specified gateway. From source Site A, the network interfaces `enp130s0f0` and `enp130s0f1` are configured with addresses in the `30.30.214.0/24` and the `32.32.214.0/24` ranges, respectively. We can configure a route-interface file for each to direct packets for the remote networks over the specified gateway and interface.

`route-enp130s0f0` contents → `31.31.0.0/16 via 30.30.214.252 dev enp130s0f0`

`route-enp130s0f1` contents → `33.33.0.0/16 via 32.32.214.252 dev enp130s0f1`

Packets intended for the remote network 31.31.214.0/24 are directed through the next hop address at gateway IP 30.30.214.252. And similarly for packets destined for 33.33.214.0/24.

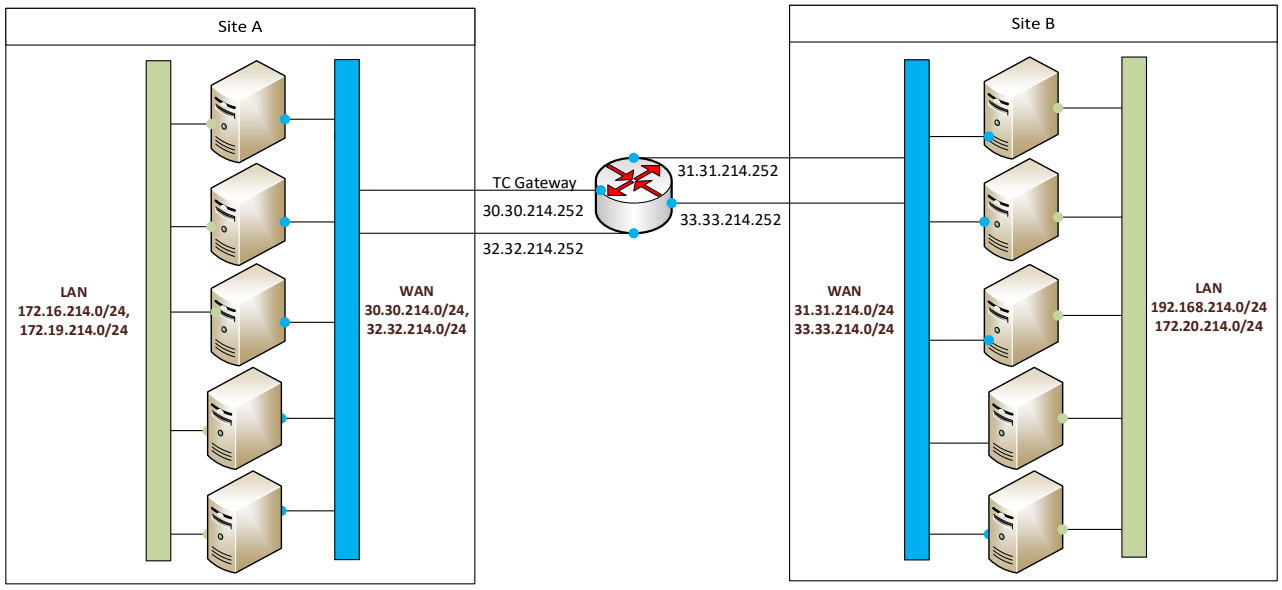


Figure 11 Example WAN topology for PowerFlex replication.

The details of static route configuration will vary with your operating system / hypervisor and overall network architecture, but the general principle is the same.

## 14.4 MTU and Jumbo frames

MTU must be set properly on the inter-SDR network interfaces in order to match the WAN link configuration. In many cases, this will be 1500. This is especially important to remember if jumbo frames are enabled on all local networks as a performance enhancement. IP fragmentation when MTU does not match the WAN configuration will result in diminished replication performance. Depending on the hardware configuration, MTU mismatches can result in packets being dropped altogether when reaching an interface. Therefore, in all cases, the MTU of the WAN must be both known and tested.

# 15 Dynamic Routing Considerations

In large leaf-spine environments consisting of hundreds of nodes, the network infrastructure may be required to dynamically route PowerFlex traffic.

A central objective to routing PowerFlex traffic is to reduce the convergence time of the routing protocol. When a component or link fails, the router or switch must detect the failure; the routing protocol must propagate the changes to the other routers; then each router or switch must re-calculate the route to each destination node. If the network is configured correctly, this process can happen in less than 300 milliseconds: fast enough to maintain MDM cluster stability.

If, during extreme congestion or network failure, the convergence time exceeds 400 milliseconds, the MDM cluster may fail over to a secondary MDM. The system will continue to operate, and I/O will continue, if the MDM fails over, nevertheless **300 milliseconds is the target to maintain maximum system stability**. Timeout values for other system component communication mechanisms are much higher, so the system should be designed for the most demanding timeout requirements: those of the MDMs.

For the fastest possible convergence time, standard best practices apply. This means conforming to all network vendor best practices designed to achieve that end, including the absence of underpowered routers (weak links) that prevent rapid convergence.

Convergence time is insufficient in every tested network vendor's default OSPF or BGP configuration. **Every routing protocol deployment, irrespective of network vendor, must include performance tweaks to minimize convergence time.** These tweaks include the use of Bidirectional Forwarding Detection (BFD) and the adjustment of failure-related timing mechanisms.

OSPF and BGP have both been tested with PowerFlex. PowerFlex is known to function without errors during link and device failures when routing protocols and networking devices are configured properly. However, **OSPF is recommended over BGP**. This recommendation is supported by test results that indicate OSPF converges faster than BGP when both are configured optimally for fast convergence.

## 15.1 Bidirectional Forwarding Detection (BFD)

Regardless of the choice of routing protocol (OSPF or BGP), the use of Bidirectional Forwarding Detection (BFD) is required. BFD reduces the overhead associated with protocol-native hello timers, allowing link failures to be detected quickly. BFD provides faster failure detection than native protocol hello timers for a number of reasons including reduction in router CPU and bandwidth utilization. **BFD is therefore strongly recommended over aggressive protocol hello timers**

PowerFlex is stable during network failovers when it is deployed with BFD and optimized OSPF and BGP routing. Sub-second failure detection must be enabled with BFD.

For a network to converge, the event must be detected, propagated to other routers, processed by the routers, and the routing information base (RIB) or Forwarding Information Base (FIB) must be updated. All these steps must be performed for the routing protocol to converge, and they should all complete in less than 300 milliseconds.

In tests using Cisco 9000 series switches a **BFD hold down timer of 150 milliseconds** was sufficient. The configuration for a 150 millisecond hold down timer consisted of 50 millisecond transmission intervals, with a 50 millisecond min_rx and a multiplier of 3. The PowerFlex recommendation is to use a maximum hold down timer of 150 milliseconds. If your switch vendor supports BFD hold down timers of less than 150 milliseconds,

the shortest achievable hold down timer is preferred. BFD should be enabled in asynchronous mode when possible.

In environments using Cisco vPC (MLAG), **BFD should also be enabled on all routed interfaces and all host-facing interfaces running First Hop Redundancy Protocol (FHRP).**

```
feature bfd

hsrp bfd all-interfaces

interface Vlan<num>
no shutdown
no ip redirects
ip address 192.168.103.2/24
no ipv6 redirects
hsrp version 2
hsrp 103
authentication text Vce12345
preempt
priority 110
ip 192.168.103.1

router ospf 1
  bfd
  bfd all-interfaces strict-mode

interface eth <x/x> / vlan <num> / Po <num>
 bfd interval 50 min_rx 50 multiplier 3
```

Figure 12 An example of a BFD configuration on a Cisco switch using an Aggregation – Access/Spine Leaf topology. BFD is configured with a hold down timer of 150 milliseconds (the interval is 50 microseconds; the multiplier is 3). OSPF on interface port-channel51 and HSRP on interface Vlan30 are both configured as a client of BFD.

```
bfd ipv4 interval 50 min_rx 50 multiplier 3


interface Vlan30
  bfd interval 50 min_rx 50 multiplier 3
  no bfd echo
  vrrp 1
    vrrp bfd 30.30.30.124

interface port-channel49
  no bfd echo
  bfd per-link

interface port-channel51
  no bfd echo
  bfd per-link
router ospf 100
  bfd
```

Figure 13 An example of a Dell BFD configuration in an Aggregation – Access topology. BFD is configured with a hold down timer of 150 milliseconds (the interval is 50 microseconds; the multiplier is 3). OSPF on interface port-channel51 and VRRP on interface Vlan30 are both configured as a client of BFD.

Note the following about these configurations:

- For port-channel interface, BFD per-link must be enabled.
- IP redirect must be disabled for BFD. (An override to ensure that BFD works)
- FHRP is only required for Access/Aggregation topology

## 15.2 Physical Link Configuration

Timers involved with link failures are candidates for tuning. Link down and interface down event detection and handling varies by network vendor and product line. **On Cisco Nexus switches, "`carrier-delay`" timer should be set to 100 milliseconds on each SVI interface, and "`link debounce`" timer should be set to 500 milliseconds on each physical interface.**

Carrier delay (`carrier-delay`) is a timer on the switch. It is applicable to an SVI interface. Carrier delay represents the amount of time the switch should wait before it notifies the application when a link failure is detected. Carrier delay is used to prevent flapping event notification in unstable networks. In modern leaf-spine environments, all links should be configured as point-to-point, providing a stable network. The recommended value for an SVI interface carrying PowerFlex traffic is 100 milliseconds.

Debounce (`link debounce`) is a timer that delays link-down notification in firmware. It is applicable to a physical interface. Debounce is similar to carrier delay, but it is applicable to physical interfaces, rather than logical interfaces, and is used for link down notifications only. Traffic is stopped during the wait period. A nonzero link debounce setting can affect the convergence of routing protocols. The recommended value for a link debounce timer is 500 milliseconds for a physical interface carrying PowerFlex traffic.

```
interface vlan <num>
 carrier-delay msec 100

interface eth <x/x>
  link debounce time 500
```

## 15.3 ECMP

**The use of Equal-Cost Multi-Path Routing (ECMP) is required.** ECMP distributes traffic evenly between leaf and spine switches, and provides high availability using redundant leaf to spine network links. ECMP is analogous to MLAG, but operates over layer 3 (IP), rather than over Ethernet.

ECMP is on by default with OSPF on Cisco Nexus switches. It is not on by default with BGP on Cisco Nexus switches, so it must be enabled manually. The ECMP hash algorithm used should be layer 3 (IP) or layer 3 and layer 4 (IP and TCP/UDP port).

## 15.4 OSPF

OSPF is the preferred routing protocol because when it is configured properly, it converges rapidly. When OSPF is used, the leaf and spine switches all reside in a single OSPF area. **To provide stable intra-MDM communication, a sub-300 millisecond convergence time is required.** On all leaf and spine switches, the OSPF interfaces should be configured as point-to-point with the OSPF process configured as a client of BFD.

This ensures that the timers are set correctly; do not vary from default. **Additionally, for L3 handoff in ToR-Agg (Access-Agg) topologies, OSPF interfaces should be configured as point-to-point.**

## 15.5 BGP

Though OSPF is preferred because it can converge faster, BGP can also be configured to converge within the required time frame.

**BGP is not configured to use ECMP on Cisco Nexus switches by default. It must be configured manually.** Both IBGP and EBGP do not support ECMP by default and must be configured . Configuration of IBGP requires a BGP route reflector and the add-path feature to fully support ECMP in a spine-and-leaf topology.

BGP can be configured in a way where each leaf and spine switch represents a different Autonomous System Number (ASN). In this configuration, each leaf has to peer with every other spine.

**Leaf and spine switches should also enable ECMP by allowing the switch to load balance across multiple BGP paths, .** On Cisco, this includes setting the "`maximum-path`" parameter to number of available paths to spine switches.

**BGP with PowerFlex requires that BFD be configured on each leaf and spine neighbor.** When using BGP, **the SDS and MDM networks are advertised by the leaf switch.**

**Leaf Configuration**

```
router bgp 100
  router-id 1.1.1.2
  address-family ipv4 unicast
    maximum-paths ibgp 3
    address-family l2vpn evpn
    maximum-paths ibgp 3

  neighbor 11.11.11.11
    bfd
    remote-as 100
    update-source loopback0
    address-family ipv4 unicast
      send-community
      send-community extended
    address-family l2vpn evpn
      send-community
      send-community extended

  vrf VxFLEX_MGMTanagement_VRF
    address-family ipv4 unicast
      maximum-paths ibgp 3
      advertise l2vpn evpn
      redistribute direct route-map ALL
```

**Spine Configuration**

```
router bgp 100
  router-id 11.11.11.11
  address-family ipv4 unicast
   maximum-paths ibgp 3
  address-family l2vpn evpn
   maximum-paths ibgp 3

  neighbor 1.1.1.1
    bfd
    remote-as 100
    update-source loopback0
    address-family ipv4 unicast
    address-family l2vpn evpn
      send-community
      send-community extended
      route-reflector-client
```

Figure 14 BGP configuration examples on a Cisco Nexus leaf switch (left) and spine switch (right). They reside in same autonomous systems (100). The "`maximum-path`" parameter is tuned to match the number of paths to be used for ECMP. (In this example, it is 3, but that may not always be the case). BFD is enabled for each leaf or spine neighbor. The leaf switch is configured to advertise the PowerFlex MDM and SDS networks

NOTE:

- On PowerFlex Rack systems that use spine-and-leaf topology, BGP is used for communication of control plane and reachability for EVPN. OSPF is used for the data plane.
- Maximum-paths allows for multiple NVE interface VTEP reachability
- IBGP is configured with the use of spines as Route-reflectors
- BGP as-path multipath-relax is not applicable due to not using EBGP

## 15.6 Leaf to Spine Bandwidth Requirements

Assuming storage media is not a performance bottleneck, calculating the amount of bandwidth required between leaf and spine switches involves determining the amount of bandwidth available from each leaf switch to the attached hosts, discounting the amount if I/O that is likely to be local to the leaf switch, then dividing the remote bandwidth requirement between each of the spine switches.

Consider a situation with two racks where each rack contains two leaf switches and 20 servers, each server has two 25 gigabit interfaces, and each of these servers is dual-homed to the two leaf switches in the rack. In this case, the downstream bandwidth from each of the leaf switches is calculated as:

$$20\ servers * 25 \frac{gigabits}{server} = 500\ gigabits$$

The downstream bandwidth requirement for each leaf switch is 500 gigabits. However, some of the traffic will be local to the pair of leaf switches, and therefore will not need to traverse the spine switches.

The amount traffic that is local to the leaf switches in the rack is determined by the number of racks in the configuration. If there are two racks, 50% of the traffic will likely be local. If there are three racks, 33% of the traffic will likely be local. If there are four racks, 25% of the traffic is likely to be local, and so on. Stated differently, the proportion of I/O that is likely to be remote will be:

$$remote\_ratio\ = \frac{number\_of\_racks - 1}{number\_of\_racks}$$

In this example, there are two racks, so 50% of the bandwidth is likely to be remote:

$$remote\_ratio\ = \frac{2\ total\_racks - 1\ rack}{2\ total\_racks} = 50\%$$

Given that there are two racks in this example, 50% of the bandwidth is likely to be remote. Multiply the amount of traffic expected to be remote by the downstream bandwidth of each leaf switch to find the total remote bandwidth requirement from each leaf switch:

$$per\_leaf\_requirement = 500\ gigabits * 50\%\ remote\_ratio = 250\ gigabits$$

250 gigabits of bandwidth is required between the leaf switches in this example with 25GbE networks. However, this bandwidth will be distributed between spine switches, so an additional calculation is required.

To find the upstream requirements to each spine switch from each leaf switch, divide the remote bandwidth requirement by the number of spine switches, since remote load is balanced between the spine switches.

$$per\_leaf\_to\_spine\_requirement = \frac{per\_leaf\_requirement}{number\_of\_spine\_switches}$$

In this example, each leaf switch is expected to demand 250 gigabits of remote bandwidth through the mesh of spine switches. Since this load will be distributed among the spine switches (assume there are two), the total bandwidth between each leaf and spine is calculated as:

$$per\_leaf\_to\_spine\_requirement\ = \frac{250\ gigabits}{2\ spine\ switches} = 125\ \frac{gigabits}{spine\ switch}$$

Therefore, for a nonblocking topology, two 100 gigabit connections for a total of 200 gigabits is sufficient bandwidth between each leaf and spine switch. Alternatively, one could divide 125Gb/s among four 40 gigabit connections.

The equation to determine the amount of bandwidth needed from each leaf switch to each spine switch can be summarized as:

$$\frac{downstream\_bandwidth\_requirement\ * ((number\_of\_racks - 1)\ /number\_of\_racks)}{number\_of\_spine\_switches}$$

Note: in systems where replication is implemented, these calculations must accommodate the additional back-end replication storage traffic. This will likely double the requirements in these examples – four 25 gigabit interfaces to the leaf switches, etc.

## 15.7 FHRP Engine

For routed access architectures with Cisco vPC and IP-level redundancy on the nodes, Dell recommends using FHRP for the node default gateway. This allows the default gateway to fail over to the other leaf switch in the event of leaf switch failure. The FHRP engine will vary by switch vendor used. When using Cisco architecture HSRP is used. For Dell switches VRRP is used.

**Aggregation Switch 1**

```
interface Vlan103
no shutdown
mtu 9216
no ip redirects
ip address 192.168.103.2/24
no ipv6 redirects
hsrp version 2
hsrp 103
authentication text <text>
preempt
priority <value>
ip 192.168.103.1
```

**Aggregation Switch 2**

```
interface Vlan103
no shutdown
mtu 9216
no ip redirects
ip address 192.168.103.3/24
no ipv6 redirects
hsrp version 2
hsrp 103
authentication text <text>
preempt
ip 192.168.103.1
```

Figure 15 A FHRP engine configuration example on a pair of Cisco Nexus Aggregation switches. . The active vPC peer should act as the FHRP primary while the backup vPC peer should act as the FHRP secondary.

# 16 VMware Considerations

Though network connections are virtualized in ESXi, the same principles of physical network layout described in this document apply. Specifically, this means that MLAG should be avoided on links carrying MDM traffic unless a Dell EMC PowerFlex representative has been consulted.

It is helpful to think of physical network from the perspective of the network stack on the virtual machine running the MDM or SDS, or the network stack in use by the SDC in the VMkernel. Considering the needs of the guest or host level network stack, then applying it to the physical network can inform decisions about the virtual switch layout.

Note: in version 3.5 native asynchronous replication is not yet supported in VMware-based hyperconverged systems. Therefore, the IP and throughput considerations noted above for Linux-based systems do not immediately apply in this case. But if users wish to plan forward, the additional throughput considerations outlined in section 7.2.3 should be accounted for.

## 16.1 IP-level Redundancy

**When network link redundancy is provided using a dual subnet configuration, two separate virtual switches are needed.** This is required because each virtual switch has its own physical uplink port. When PowerFlex is run in hyper-converged mode, this configuration has 3 interfaces: VMkernel for the SDC, VM network for the SDS, and uplink for physical network access. PowerFlex natively supports installation in this mode.

## 16.2 LAG and MLAG

**The use of the distributed virtual switch is required when LAG or MLAG is used.** The standard virtual switch does not support LACP and is therefore not recommended. When LAG or MLAG is used, the bonding is done on physical uplink ports.

PowerFlex installation using the vSphere plugin does not natively support LAG or MLAG installation. Instead, it can be created prior to the PowerFlex deployment and selected during the installation process.

If a node running an SDS or SDC has aggregated links to a switch, the hash mode on the physical uplink ports should be configured to use “Source and destination IP address” or “Source and destination IP address and TCP/UDP port”.

We recommend using this only as a second level of redundancy, if desired.

## 16.3 SDC

The SDC is a kernel driver for ESXi that implements the PowerFlex storage client. Since it runs in the ESXi kernel, it uses one or more VMkernel ports for communication with the other PowerFlex components. We repeat our general recommendation to implement native IP-level redundancy, which , in this case, means each VMkernel port is mapped to a distinct physical port. If a second level of redundancy is desired, LAG or MLAG can be implemented on the distributed switch layer in addition to IP-level redundancy.

## 16.4 SDS

The SDS is deployed as a part of the virtual storage appliance (SVM) on ESXi. Again, our recommended implementation uses native IP-level redundancy, with each subnet assigned to its own virtual switch and physical uplink port. If a second level of redundancy is desired, LAG or MLAG can be implemented on the distributed switch layer in addition to IP-level redundancy.

## 16.5 MDM

The MDM is deployed as a part of the virtual storage appliance (SVM) on ESXi. The used of IP-level redundancy is strongly recommended. **A single MDM should therefore use two or more separate virtual switches.**

# 17 Virtualized and Software-defined Networking

We shall have more to say in a future update. We make these brief notes to clear misunderstandings about SDN support in general.

## 17.1 Cisco ACI

We do not have direct or full support for PowerFlex over Cisco ACI. In particular, we do not support backend storage traffic over Cisco ACI. However, we can support it in a dual network extension, where frontend customer traffic flows over the ACI fabric.

## 17.2 Cisco NX-OS

We support VxLAN EVPN Leaf Spine fabric with NX-OS Standalone software.

# 18 Validation Methods

## 18.1 PowerFlex Native Tools

There are two main built-in tools that monitor network performance:

1. SDS Network Test
2. SDS Network Latency Meter Test

### 18.1.1 SDS Network Test

Usage of the SDS network test, "`start_sds_network_test`", is covered in the Dell EMC PowerFlex v3.5 CLI Reference Guide. To fetch the results after it is run, use the "`query_sds_network_test_results`" command.

It is important to note that the `parallel_messages` and `network_test_size_gb` options should be set so that they are at least 2x larger than the maximum network bandwidth of the link over which the test is run. For example: a single 10GbE NIC = 1250 megabytes * 2 = 2500 megabytes, or 3 gigabits rounded up. In this case, the command should use the parameter "`--network_test_size_gb 3`" This will ensure that enough bandwidth is sent out on the network to get a consistent test result. For 25GbE network configurations, a single 25GbE NIC = 3125 megabytes * 2 = 6250 megabytes, or 6 gigabits. In that case, the command should include "`--network_test_size_gb 6`".

The parallel message size should be equal to the total number of cores in your system, with a maximum configuration of 16.

**Note:** This test should be run on each SDS for each configured SDS network.

**Example Output:**

```
scli --start_sds_network_test --sds_ip 10.248.0.23 --network_test_size_gb 8 --parallel_messages 8
Network testing successfully started.

scli --query_sds_network_test_results --sds_ip 10.248.0.23SDS with IP
10.248.0.23 returned information on 7 SDSs
    SDS 6bfc235100000000 10.248.0.24 bandwidth 2.4 GB (2474 MB) per-second
    SDS 6bfc235200000001 10.248.0.25 bandwidth 3.5 GB (3592 MB) per-second
    SDS 6bfc235400000003 10.248.0.26 bandwidth 2.5 GB (2592 MB) per-second
    SDS 6bfc235500000004 10.248.0.28 bandwidth 3.0 GB (3045 MB) per-second
    SDS 6bfc235600000005 10.248.0.30 bandwidth 3.2 GB (3316 MB) per-second
    SDS 6bfc235700000006 10.248.0.27 bandwidth 3.0 GB (3056 MB) per-second
    SDS 6bfc235800000007 10.248.0.29 bandwidth 2.6 GB (2617 MB) per-second
```

In the example above, you can see the network performance from the SDS you are testing to every other SDS on the network segment. Ensure that the speed per second is close to the expected performance of your network configuration.

### 18.1.2 SDS Network Latency Meter Test

The "`query_network_latency_meters`" command can be used to show the average network latency between SDS components. Low latency between SDS components is crucial for good write performance. When running this test, look for outliers and latency higher than a few hundred microseconds when 10 gigabit or better network connectivity is used.

**Note:** this should be run from each SDS *and* over each SDS network.

**Example Output:**

```
scli --query_network_latency_meters --sds_ip 10.248.0.23
SDS with IP 10.248.0.23 returned information on 7 SDSs

SDS 10.248.0.24
  Average IO size: 8.0 KB (8192 Bytes)
  Average latency (micro seconds): 231

SDS 10.248.0.25
  Average IO size: 40.0 KB (40960 Bytes)
  Average latency (micro seconds): 368

SDS 10.248.0.26
  Average IO size: 38.0 KB (38912 Bytes)
  Average latency (micro seconds): 315

SDS 10.248.0.28
  Average IO size: 5.0 KB (5120 Bytes)
  Average latency (micro seconds): 250

SDS 10.248.0.30
  Average IO size: 1.0 KB (1024 Bytes)
  Average latency (micro seconds): 211

SDS 10.248.0.27
  Average IO size: 9.0 KB (9216 Bytes)
  Average latency (micro seconds): 252

SDS 10.248.0.29
  Average IO size: 66.0 KB (67584 Bytes)
  Average latency (micro seconds): 418
```

## 18.2 Iperf, NetPerf, and Tracepath

**NOTE**: Iperf and NetPerf should be used to validate your network before configuring PowerFlex. If you identify issues with Iperf or NetPerf, there may be network issues that need to be investigated. If you do not see issues with Iperf/NetPerf, use the PowerFlex internal validation tools for additional and more accurate validation.

**Iperf** is a traffic generation tool, which can be used to measure the maximum possible bandwidth on IP networks. The Iperf feature set allows for tuning of various parameters and reports on bandwidth, loss, and

other measurements. When Iperf is used, it should be run with multiple parallel client threads. Eight threads per IP socket is a good choice.

**NetPerf** is a benchmark that can be used to measure the performance of many different types of networking. It provides tests for both unidirectional throughput, and end-to-end latency.

The Linux "`tracepath`" command can be used to discover MTU sizes along a path.

## 18.3 Network Monitoring

It is important to monitor the health of your network to identify any issues that are preventing your network for operating at optimal capacity, and to safeguard from network performance degradation. There are a number of network monitoring tools available for use on the market, which offer many different feature sets.

Dell Technologies recommends monitoring the following areas:

- Input and output traffic
- Errors, discards, and overruns
- Physical port status

## 18.4 Network Troubleshooting Basics

- Verify connectivity end-to-end between SDSs and SDCs using ping
- Test connectivity between components in both directions
- SDS and MDM communication should not exceed 1 millisecond network-only round-trip time.
- Verify round-trip latency between components using ping
- Check for port errors, discards, and overruns on the switch side
- Verify PowerFlex nodes are up
- Verify PowerFlex processes are installed and running on all nodes
- Check MTU across all switches and servers, especially if using jumbo frames
- Verify that MTU for the site-to-site SDR communication is adequate to the WAN
- Verify the static routing configuration for site-to-site SDR communication and test end-to-end connectivity over the WAN
- Prefer 25 gigabit or greater Ethernet in lieu of 10 gigabit Ethernet when possible
- Check for NIC errors, high NIC overrun rates (> 2%), and dropped packets in the OS event logs
- Check for IP addresses without a valid NIC association
- Verify the network ports needed by PowerFlex are not blocked by the network or the node
- Check for packet loss on the OS running PowerFlex using event logs or OS network commands
- Verify no other applications running on the node are attempting to use TCP ports required by PowerFlex
- Set all NICs to full duplex, with auto negotiation on, and the maximum speed supported by your network

- Check PowerFlex native tool test outputs
- Check for RAID controller misconfiguration (this is not network related, but it is a common performance problem)
- If you have a problem, collect the logs as soon as you can before they are over-written
- Additional troubleshooting, log collection information, and an FAQ is available in Troubleshoot and Maintain Dell EMC PowerFlex v3.5 and PowerFlex v3.5 Log Collection Technical Notes.

# 19 Conclusion

The selected deployment option, network topology, performance requirements, Ethernet, dynamic IP routing, and validation methods, all factor into a robust and sustainable network design. Dell EMC PowerFlex clusters can scale up to 1024 nodes containing a variety of node types, storage media, and deployment configurations, so one should size the network installation for future growth. The fact that PowerFlex can be deployed in a hyper-converged mode where compute and storage reside on the same set of nodes, or in a two-layer mode where storage and compute resources are separate affect your decisions as well. To achieve immense performance, scalability, and flexibility, the network must be designed to account for the needs of the business. Following the principles and recommendations in this guide will result in a resilient, massively scalable, and high-performing block storage infrastructure.