

DELL EMC POWERSCALE

CONSIDERATIONS & BEST PRACTICES FOR LARGE CLUSTERS

Abstract

This paper describes best practices for installing, configuring and managing a large Dell EMC PowerScale cluster.

February 2021

Revisions

Version	Date	Comment
1.0	November 2017	Updated for OneFS 8.1.1
2.0	February 2019	Updated for OneFS 8.1.3
3.0	April 2019	Updated for OneFS 8.2
4.0	August 2019	Updated for OneFS 8.2.1
5.0	December 2019	Updated for OneFS 8.2.2
6.0	June 2020	Updated for OneFS 9.0
7.0	September 2020	Updated for OneFS 9.1

Acknowledgements

This paper was produced by the following:

Author: Nick Trimbee

The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

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

Copyright © Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.

TABLE OF CONTENTS

Intended Audience 4

Cluster Definitions 4

Considerations for Planning and Designing a Large Cluster 5

 Cluster Architecture 6

 Single Large Cluster 6

 Storage Pod 7

 Workload Analysis 8

How does the application work? 8

 What is the disk latency? 9

 How much CPU utilization? 9

Large Cluster Health Check 10

 Hardware Considerations 10

Cluster Administration & Management Considerations 15

 Cluster Capacity Considerations 18

 OneFS Software Considerations 21

 Cluster Composition and Group State 21

 Layout, Protection, and Failure Domains 28

 Tiers 29

 Failure Domains & Neighborhoods 29

 File Layout and Directory Structure 31

Data Layout and Tiering Recommendations 32

Multi-tenant Recommendations 33

Job Engine Recommendations 34

 Data Availability, Protection & Disaster Recovery Considerations 36

 Replication Recommendations 37

Best Practices Checklist 38

Summary 39

Intended Audience

This paper outlines considerations and best practices for deploying and managing a large Dell EMC PowerScale cluster. It also offers configuration and tuning recommendations to help achieve optimal performance for different workloads. This paper does not intend to provide a comprehensive background to the OneFS architecture.

 Please refer to the [OneFS Technical Overview](#) white paper for further details on the OneFS architecture.

The target audience for this white paper is anyone designing and deploying a large OneFS powered clustered storage environment. It is assumed that the reader has an understanding and working knowledge of the OneFS components, architecture, commands and features.

 More information on OneFS commands and feature configuration is available in the [OneFS Administration Guide](#).

Cluster Definitions

Depending on who you ask, you'll likely receive several definitions of exactly what constitutes a large cluster. Criteria often include:

- Capacity and file count-based definitions
- Layout based answers
- Node count-based focus

Some notable landmarks that are reached as the cluster node count increases include:

Node Count	Description of attribute
20	The largest number of nodes (stripe width) that OneFS can write data and parity blocks across (stripe width). Point at which Isilon Gen6 nodes split into two neighborhoods.
32	Larger, modular enterprise class switch for Ethernet backend clusters is often deployed (>32 ports).
40	Point at which OneFS automatically divides into two PowerScale (and Isilon Gen5 earlier) neighborhoods, and Isilon Gen6 clusters achieve chassis-level redundancy (four neighborhoods).
48	Larger, modular enterprise class switch for Infiniband backend clusters is required (>48 ports)
64	Historically recommended maximum cluster size
80	No InsightIQ support, CELOG alerting challenges, WebUI and CLI list processing and reporting become cumbersome.
144	OneFS 8.1.x and earlier maximum supported cluster node count
252	OneFS 8.2 and later maximum supported cluster node count

Figure 1: Cluster node count growth events

For the purposes of this paper, we'll focus on node count as the cluster size criteria and consider the following definitions:

Definition	PowerScale Description	Isilon Gen6 Description
Small cluster	Between 3 and 32 nodes	Between 1 and 8 chassis
Medium cluster	Between 32 and 48	Between 9 and 12 chassis
Large cluster	Between 48 and 144 nodes	Between 13 and 36 chassis
Extra-large cluster	Between 144 and 252 nodes	Between 20 and 64 chassis

Figure 2: Cluster size definitions

Prior to OneFS 8.0, the recommendation was for a maximum cluster size of around 64 nodes based on balancing customer experience with the manageability of extra-large clusters, the risk profile associated with the size of the fault domain that represents for their business, and the ease and simplicity of a single cluster. However, since then, OneFS 8 and later releases have seen considerable backend network infrastructure enhancements removing this 64-node max recommendation and providing cluster stability up to the current supported maximum of 252 nodes per cluster in OneFS 8.2 and later.

One of the significant developments in cluster scaling has been the introduction of Ethernet as a cluster's backend network in Isilon Gen6 and PowerScale hardware. However, it is still possible to use Isilon Gen6 nodes with an Infiniband backend for compatibility with previous generations of nodes. This allows legacy clusters to be augmented with the new generation of hardware.

① When provisioning an all-new cluster comprised of Isilon Gen6 nodes exclusively, 40Gb Ethernet is highly encouraged for the backend interconnect network. Additionally, Ethernet backend is strongly recommended for extra-large clusters, and configurations using Dell switches configured in a leaf-spine topology are supported all the way up to 252 nodes.

Considerations for Planning and Designing a Large Cluster

When it comes to architecting and scaling large OneFS powered clusters, there are some key tenets to bear in mind. These include:

- Strive for simplicity
- Plan ahead
- Just because you can doesn't necessarily mean you should

Distributed systems tend to be complex by definition, and this is amplified at scale. OneFS does a good job of simplifying cluster administration and management, but a solid architectural design and growth plan is key. Because of its single, massive volume and namespace, a OneFS viewed by many as a sort of 'storage Swiss army knife'. Left unchecked, this methodology can result in unnecessary complexities as a cluster scales. As such, decision making that favors simplicity is key.

Despite OneFS' extensibility, allowing a system to simply grow organically into a large cluster often results in various levels of technical debt. In the worst case, some issues may have grown so large that it becomes impossible to correct the underlying cause. This is particularly true in instances where a small cluster is initially purchased for an archive or low performance workload and with a bias towards cost optimized storage. As the administrators realize how simple and versatile their clustered storage environment is, more applications and workflows are migrated to OneFS. This kind of haphazard growth, such as morphing from a low-powered, near-line platform into something larger and more performant, can lead to all manner of scaling challenges. However, compromises, living with things, or fixing issues that could have been avoided can usually be mitigated by starting with a scalable architecture, workflow and expansion plan.

Starting with a defined architecture, sizing and expansion plan is key. What do you anticipate the cluster, workloads, and client access levels will look like in six months, one year, three years, or five years? How will you accommodate the following as the cluster scales?

- Contiguous rack space for expansion
- Sufficient power & cooling
- Network infrastructure
- Backend switch capacity
- Availability SLAs
- Serviceability and spares plan
- Backup and DR plans
- Mixed protocols
- Security, access control, authentication services, and audit
- Regulatory compliance and security mandates
- Multi-tenancy and separation
- Bandwidth isolation – client I/O, replication, etc.
- Application and workflow expansion

Cluster Architecture

There are two distinct paths to pursue when initially designing a clustered storage architecture for a large and/or rapidly growing environment - particularly one that includes a performance workload element to it. These are:

- Single Large Cluster
- Storage Pod Architecture

Single Large Cluster

A single large, or extra-large, cluster is often deployed to support a wide variety of workloads and their requisite protocols and performance profiles – from primary to archive - within a single, scalable volume and namespace. This approach, referred to as a 'data lake architecture', usually involves more than one style of node.

OneFS can support up to fifty separate tenants in a single cluster, each with their own subnet, routing, DNS, and security infrastructure. OneFS provides the ability to separate data layout with SmartPools, export and share level isolation, granular authentication and access control with Access Zones, and network partitioning with SmartConnect, subnets, and VLANs.

Furthermore, analytics workloads can easily be run against the datasets in a single location and without the need for additional storage and data replication and migration.

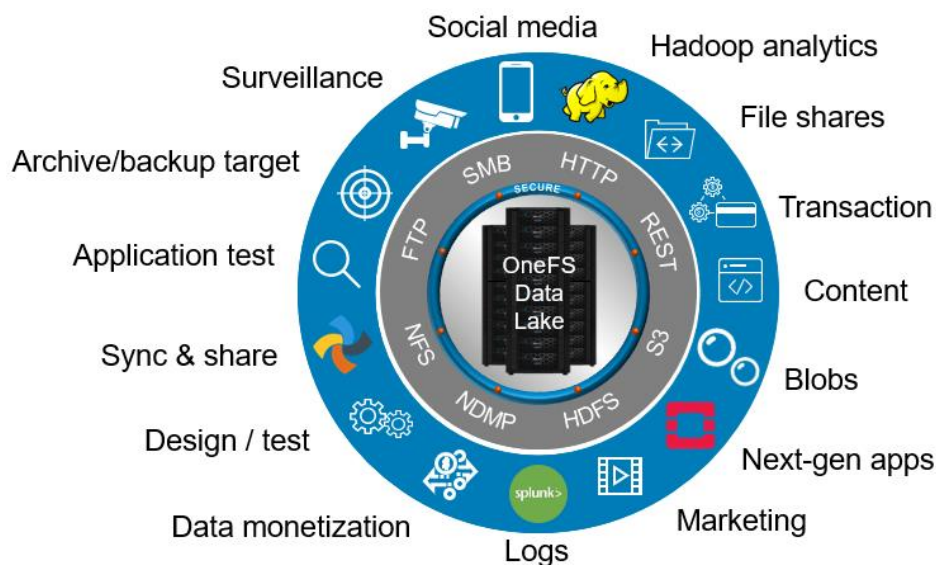


Figure 3: OneFS Data Lake

For the right combination of workloads, the data lake architecture has many favorable efficiencies of scale and centralized administration.

Another use case for large clusters is in a single workflow deployment, for example as the content repository for the asset management layer of a content delivery workflow. This is a considerably more predictable, and hence simpler to architect, environment than the data lake.

Often, as in the case of a MAM for streaming playout, a single node type is deployed. The I/O profile is typically heavily biased towards streaming reads and metadata reads, with a smaller portion of writes for ingest.

There are trade-offs to be aware of as cluster size increases into the extra-large cluster scale. The larger the node count, the more components are involved, which increases the likelihood of a hardware failure. When the infrastructure becomes large and complex enough, there's more often than not a drive failing or a node in an otherwise degraded state. At this point, the cluster can be in a state of flux such that composition, or group, changes and drive rebuilds/data re-protection operations will occur frequently enough that they can start to significantly impact the workflow.

Higher levels of protection are required for large clusters, which has a direct impact on capacity utilization. Also, cluster maintenance becomes harder to schedule since many workflows, often with varying availability SLAs, need to be accommodated.

Additional administrative shortcomings that also need to be considered when planning on an extra-large cluster include that InsightIQ only supports monitoring clusters of up to eighty nodes and the OneFS Cluster Event Log (CELOG) and some of the cluster WebUI and CLI tools can prove challenging at an extra-large cluster scale.

That said, there can be wisdom in architecting a clustered NAS environment into smaller buckets and thereby managing risk for the business vs putting the ‘all eggs in one basket’. When contemplating the merits of an extra-large cluster, also consider:

- Performance management,
- Risk management
- Accurate workflow sizing
- Complexity management.

Storage Pod

An alternative approach for HPC, and high-IOPS workloads is the Storage Pod architecture. Here, design considerations for new clusters revolve around multiple smaller homogenous clusters, with each cluster itself acting as a fault domain – in contrast to the monolithic extra-large cluster described above.

Pod clusters can easily be tailored to the individual demands of workloads as necessary. Optimizations per storage pod can include size of SSDs, drive protection levels, data services, availability SLAs, etc. In addition, smaller clusters greatly reduce the frequency and impact of drive failures and their subsequent rebuild operations. This, coupled with the ability to more easily schedule maintenance, manage smaller datasets, simplify DR processes, etc, can all help alleviate the administrative overhead for a cluster.

A Pod infrastructure can be architected per application, workload, similar I/O type (ie. streaming reads), project, tenant (ie. business unit), availability SLA, etc. This pod approach has been successfully adopted by a number of large customers in industries such as semiconductor, automotive, life sciences, and others with demanding performance workloads.

This Pod architecture model can also fit well for global organizations, where a cluster is deployed per region or availability zone. An extra-large cluster architecture can be usefully deployed in conjunction with Pod clusters to act as a centralized disaster recovery target, utilizing a hub and spoke replication topology. Since the centralized DR cluster will be handling only predictable levels of replication traffic, it can be architected using capacity-biased nodes.



Figure 4: Pod Architecture

① Before embarking upon either a data lake or Pod architectural design, it is important to undertake a thorough analysis of the workloads and applications that the cluster(s) will be supporting.

Workload Analysis

Despite the flexibility offered by the data lake concept, not all unstructured data workloads or applications are suitable for a large OneFS powered cluster. Each application or workload that is under consideration for deployment or migration to a cluster should be evaluated carefully. Workload analysis involves reviewing the ecosystem of an application for its suitability. This requires an understanding of the configuration and limitations of the infrastructure, how clients see it, where data lives within it, and the application or use cases in order to determine:

- How the application works?
- How users interact with the application?
- What is the network topology?
- What are the workload-specific metrics for networking protocols, drive I/O, and CPU & memory usage?

How does the application work?

First, determine whether an application or workflow has a unique or unbalanced dataset, determine its core characteristics. For example, does it include any of the following?

- Broad and deeply nested directory trees with few files per directory
- Shallow nested directories with large numbers of files per directory
- Massive file counts (billions)
- Large files (TBs in size)
- Flat files (e.g. VMware VMDKs or relational databases)

Next, determine how the application utilizes the stored data. For example, does it involve:

- Heavy metadata reads and writes to the dataset
- Moderate metadata writes and heavy metadata reads.
- Are heavy or light data reads and/or writes required?
- Are the data reads and writes more or less random or sequential?
- Are the writes creates or appends?

If the application or workload is latency-sensitive, are there expected timeouts for data requests built into it and can they be tuned? Are there other external applications from this application that might cause latency problems, such as FindFirstFile or FindNextFile crawls to do repetitive work that is not well-suited to NAS use and needs further investigation?

If an application can benefit from caching, how much of the unique dataset will be read once and then reread and over what periods of time (hourly, daily, weekly, and so on.) This can help in appropriately sizing a cluster's L2 and L3 cache requirements.

How do users interact with the application?

Figuring out how users interact with an application is often more challenging to profile. In order to understand what levels performance users are accustomed to and what they are expecting, the following will need to be determined:

- If users will interact with the application through direct requests and responses from a flat data structure.
- If there are efficient parallelized requests, or inefficient serialized requests, to derive a result.

An example of the latter would be a CAD application that needs to load 20,000 objects from storage before a drawing can be rendered on the user's display.

What does the network topology look like?

Next, comprehensively catalog the network topology in its entirety. Maps and visualizations can help identify and resolve many issues.

- Conduct a performance study using a utility such as the 'iperf' CLI tool for network performance measurement.
- For a LAN, itemize gear models, speeds, feeds, maximum transmission units (MTUs) per link, layer two and three routing, and expected latencies (confirm).
- For a WAN, itemize providers, topologies, rate limits and guarantees, direct versus indirect pathways, and perform an 'iperf' study.

Review the application or workload's change control process. Do network and storage teams have clear lines of communication and responsibility? Look at the application's prior issue log and interview the administrators.

What is the application's performance profile?

Storage stack performance can be determined by measuring 'normal' performance levels over time and determining how to recognize when it deviates from this mean. Each cluster has its own unique configuration, dataset and workload, so will yield a unique result from its ecosystem. This effect is amplified at large and extra-large cluster scale, indicating the need for diligent and detailed monitoring and observation.

What are the application's I/O requirements?

Investigate the I/O data rates per node, per node pool, and per network pool, as applicable. Measure and understand the I/O protocol read and write rates and mix as a whole in the workload.

- Use the `'isi statistics protocol --nodes all --top --orderby=timeavg'` command to display performance statistics by protocol.
- Understand the per-protocol breakouts of the client requests—in particular, the read, write, getattr, setattr, open, close, create, and delete operations.

What is the disk latency?

Investigating and measuring the impact of non-cached workflow and transfer rates (xfers) to disks will help lead to an understanding of how a unique dataset will deliver a particular result.

- Use the `isi statistics drive --nodes all --top --long --orderby timeavg` command to display performance statistics by drive, ordered by OpsIn and OpsOut values. Note that this command is not measuring physical disk input/output operations per second (IOPS); it measures software transfer commands to storage only. Disks manage their own physical ordering of these requests, which OneFS does not see or measure in the form of physical I/O operations (IOPS). Mentally adjust the OpsIn and OpsOut fields to reflect that reality.
- It is very important to profile your workload by using at least the `isi statistics` commands. Use them to understand how an application drives the workload to and from disks.
- Unfortunately, there are no available disk transfer rate numbers that can determine how much is too much to cause performance degradation. OneFS simply does not deliver data from that deep in the storage stack to make this an easy operation. Monitor on the Busy%, Queued, TimeInAQ, and TimeAvg columns returned from the `isi statistics drive` commands to make judgments on whether your storage layer is being overwhelmed, according to your performance requirements.

How much CPU utilization?

OneFS contains several helpful tools for monitoring and measuring CPU utilization. For example, the `'isi statistics system --nodes -top'` command displays statistics for CPU performance on individual nodes.

When considering processor utilization, be aware that:

- CPU load is the result of a workload, not a leading indicator.
- CPU is an important consideration in sizing against existing equipment but is not useful for general profiling a workload.
- Writes are typically much more impactful to CPU load than reads.

When to analyze workloads?

Be sure to analyze workload performance any time performance changes, and particularly when disruption has occurred. For example, when:

- An application upgrade is performed.
- Migrating to or from a new pool.
- New functionality is enabled.

① For large or extra-large cluster architectures that are intended to support multiple workflows, a similar investigation should be performed for all the applications and workloads involved and the aggregate results examined. Contact your Dell EMC or Partner account team for assistance with analyzing your workload results.

Large Cluster Health Check

The following presents some options, both specific and general, to check for and monitor on a large cluster. The recommendation is to check these periodically, especially after a maintenance event such as an expansion or upgrade:

Core sanity check

- Are all nodes members of the group (`sysctl efs.gmp.group`)?
- If there are missing nodes, what state are they in (ie. read-only, SmartFailed or powered down)?
- Can you still write to `/ifs`?
- Do the base 'isi' commands work?

Disk usage

After a cluster comes online, note how full `/ifs` is. Check this every few hours and see if it is increasing and what the rate is. On an idle cluster, `/ifs` should not be filling up rapidly. If it does, there's a rogue application, client, or resource.

CPU and mem usage

Log into a few random nodes and check out top consumers.

- What processes are consuming CPU?
- Are they consistent or transitory?
- What about memory (RSS - resident set size specifically - shows how much memory is allocated to that process)?

If the cluster is newly formed and idle, processes should be largely idle. Processes that aren't idle are likely a bug or warrant investigation.

Log files

The `'ls -lt /var/log/'` CLI command can be an enlightening source of large cluster information:

- Which logs are growing?
- What are their contents?
- Do any of the growing logs match with processes consuming CPU?
- Are the logs filling with log spam?

Look specifically at `/var/log/messages` for errors.

Core services

Since so many services depend on the OneFS platform API for an interface, it can be helpful to run a few checks to ensure it is operational. For example:

- `wget --no-check-certificate https://localhost:8080/platform/3/cluster/nodes`
- `isi status`
- `isi job jobs list`

Similarly, check that CELOG is functioning correctly with the following command:

- `isi event list`

Hardware Considerations

Node Hardware Recommendations

A key decision for cluster performance in an environment is the type and quantity of nodes deployed. Heterogeneous clusters can be architected with a wide variety of node styles and capacities, in order to meet the needs of a varied data set and wide spectrum of workloads. These node styles encompass several hardware generations and fall loosely into four main categories or tiers.

- Extreme performance (all-flash)
- Performance
- Hybrid/Utility
- Archive

The following table illustrates these tiers, and the associated hardware generations and models:

Tier	I/O Profile	Drive Media	Node Type
Extreme Performance	High Perf, Low Latency	All-flash	F800 F810 F600 F200
Performance	Transactional I/O	SAS & SSD	H600 H5600
Hybrid / Utility	Concurrency & Streaming Throughput	SATA/SAS & SSD	H500 H400
Archive	Nearline & Deep Archive	SATA	A200 A2000

Figure 5: Cluster Hardware Tiers

① While heterogeneous clusters can easily include multiple hardware classes and configurations with a minimum of three of each, the best practice of simplicity for building large clusters holds true here too. The smaller the disparity in hardware style across the cluster, the less opportunity there is for overloading, or bullying, the more capacity-oriented nodes. Some points to consider are:

- Ensure all nodes contain at least one SSD.
- OneFS will stripe data across a maximum of 20 nodes.
- At a node pool size of 40 nodes, Isilon Gen6 hardware achieves sled, chassis and neighborhood level protection.
- When comparing equivalent Isilon Gen6 and earlier generations and types, consider the number of spindles rather than just overall capacity.

Physical and Environmental Design

Consider the physical cluster layout and environmental factors when designing and planning for a large cluster installation. These factors include:

- Redundant power supply
- Airflow and cooling
- Rackspace requirements
- Floor tile weight constraints
- Networking Requirements
- Cabling distance Limitations

Extra-large Cluster Design Example

Consider the following extra-large cluster topology, for example:

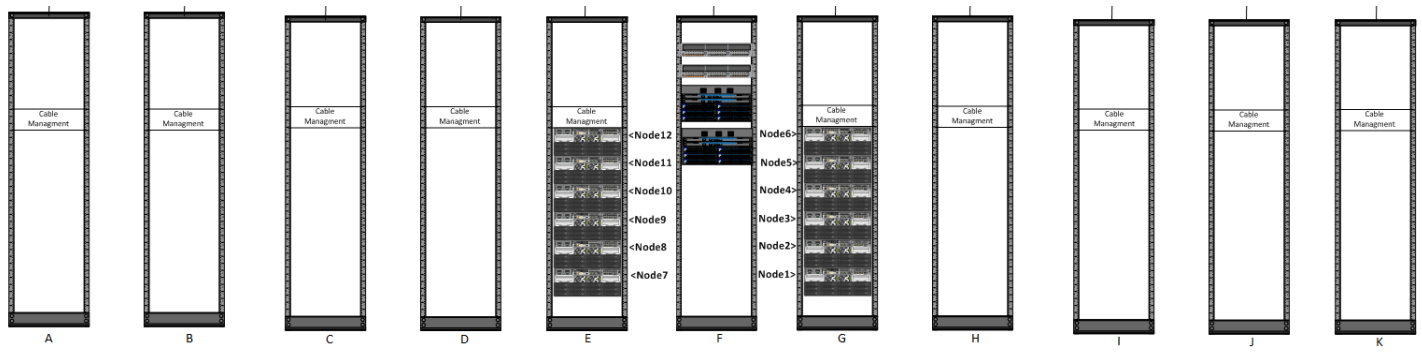


Figure 6: Extra-large Cluster Design Schematic

This contiguous eleven rack architecture is designed to scale up to ninety-six 4RU nodes as the environment grows, while keeping cable management simple and taking the considerable weight of the InfiniBand cables off the connectors as much as possible.

Best practices include:

- Pre-allocate and reserve adjacent racks in the same aisle to fully accommodate the anticipated future cluster expansion
- Reserve an empty 4RU ‘mailbox’ slot above the center of each rack for pass-through cable management.
- Dedicate the central rack in the group for the back-end and front-end switches – in this case rack F (image below).

Below, the two top Ethernet switches are for front-end connectivity and the lower two InfiniBand switches handle the cluster’s redundant back-end connections.



Figure 7: Cluster Front and Back-end Switches (Rack F Above)

The 4RU “mailbox” space is utilized for cable pass-through between node racks and the central switch rack. This allows cabling runs to be kept as short and straight as possible.



Figure 8: Rear and Side views of Rack Showing Mailbox Space and Backend Network Cabling (Rack E Above)

Excess cabling can be neatly stored in 12" service coils on a cable tray above the rack, if available, or at the side of the rack as illustrated below.

Successful large cluster infrastructures depend heavily on the proficiency of the installer and their optimizations for maintenance and future expansion.

① For Hadoop workloads, OneFS is compatible with the rack awareness feature of HDFS to provide balancing in the placement of data. Rack locality keeps the data flow internal to the rack.

Power and Cooling

In addition to available rack space and physical proximity of nodes, provision needs to be made for adequate power and cooling as the cluster expands. New generations of nodes typically deliver and increased storage density, which often magnifies the power draw and cooling requirements per rack unit.

The larger the cluster, the more disruptive downtime and reboots can be. To this end, the recommendation is for a large cluster's power supply to be fully redundant and backed up with a battery UPS and/or power generator. In the worst instance, if a cluster does lose power, the nodes are protected internally by filesystem journals which preserve any in-flight uncommitted writes. However, the time to restore power and reboot a large cluster can be considerable.

Like most data center equipment, the cooling fans in a cluster's nodes and switches pull air from the front to back of the chassis. To complement this, most data centers use a hot aisle/cold aisle rack configuration, where cool, low humidity air is supplied in the aisle at the front of each rack or cabinet either at the floor or ceiling level, and warm exhaust air is returned at ceiling level in the aisle to the rear of each rack.

Given the high power-draw and heat density of cluster hardware, some datacenters are limited in the number of nodes each rack can support. For partially filled racks, the use of blank panels to cover the front and rear of any unfilled rack units can help to efficiently direct airflow through the equipment.

The use of intelligent power distribution units (PDUs) within each rack can facilitate the remote power cycling of nodes, if desired.

① For Gen 6 hardware, where chassis depth can be a limiting factor, 2RU horizontally mounted PDUs within the rack can be used in place of vertical PDUs. If front-mounted, partial depth Ethernet switches are deployed, horizontal PDUs can be installed in the rear of the rack directly behind the switches to maximize available rack capacity.

Cabling and Networking

With copper (CX4) InfiniBand cables the maximum cable length is limited to 10 meters. After factoring in for dressing the cables to maintain some level of organization and proximity within the racks and cable trays, all the racks containing a cluster's nodes need to be in close physical proximity to each other –either in the same rack row or close by in an adjacent row.

Support for multimode fiber (SC) for InfiniBand and for Ethernet extends the cable length limitation to 150 meters. This allows nodes to be housed on separate floors or on the far side of a floor in a datacenter if necessary. While solving the floor space problem, this has the potential to introduce new administrative and management issues.

With large clusters, especially when the nodes may not be racked in a contiguous manner, having all the nodes and switches connected to serial console concentrators and remote power controllers is highly advised.

① OneFS 9.0 introduces IPMI (Intelligent Platform Management Interface) support for Isilon Gen6 and PowerScale platforms. This functionality allows out-of-band console access and remote node power control via a dedicated Ethernet management port. and is configured and managed via the 'isi ipmi' CLI command set.

However, to perform any physical administration or break/fix activity on nodes you must know where the equipment is located and have administrative resources available to access and service all locations.

As such, the following best practices are highly recommended:

- Develop and update thorough physical architectural documentation.
- Implement an intuitive cable coloring standard.
- Be fastidious and consistent about cable labeling.
- Use the appropriate length of cable for the run and create a neat 12" loop of any excess cable, secured with Velcro.
- Observe appropriate cable bend ratios, particularly with fiber cables.
- Dress cables and maintain a disciplined cable management ethos.
- Keep a detailed cluster hardware maintenance log.
- Where appropriate, maintain a 'mailbox' space for cable management.

① Disciplined cable management and labeling for ease of identification is particularly important in larger Isilon Gen6 clusters, where density of cabling is high, with each Gen6 chassis requiring up to twenty-eight cables.

The recommendation for cabling an Isilon Gen6 chassis is as follows:

- Split cabling in the middle of the chassis, between nodes 2 and 3.
- Route Ethernet and InfiniBand cables towards lower side of the chassis.
- Connect power cords for nodes 1 and 3 to PDU A and power cords for nodes 2 and 4 to PDU B.
- Bundle network cables with the AC power cords for ease of management.
- Leave enough cable slack for servicing each individual node's FRUs.

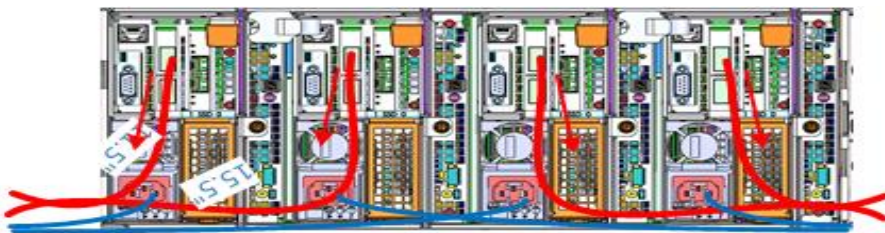


Figure 9: Gen6 Chassis Rear Illustrating Split Cabling

① Consistent and meticulous cable labelling and management is particularly important in large clusters. Isilon Gen6 chassis that employ both front and back end Ethernet networks can include up to twenty Ethernet connections per 4RU chassis.



Figure 10: Network Cable Labeling Example

① While there is no requirement that node 1 aligns with port 1 on each of the backend switches, it can certainly make cluster and switch management and troubleshooting considerably simpler. Even if exact port alignment is not possible, with large clusters, ensure that the cables are clearly labeled and connected to similar port regions on the backend switches.

Cluster Administration & Management Considerations

There are three access methods for configuring and administering a OneFS powered cluster:

- Command line interface (CLI) - either via SSH or serial console.
- Web interface (WebUI)
- RESTful platform API (PAPI)

While the Web Interface is the most intuitive, menu driven, and simple to use cluster administration method, it is also the most limited in terms of scope. The CLI has a more comprehensive set of administrative commands than the WebUI, making it a popular choice for OneFS power users.

Additionally, OneFS also provides native support for perl, python and UNIX shell scripts, which can be useful for automating management of large clusters. Creating scripts that utilize the OneFS platform API helps also avoids challenges with the CLI and WebUI in parsing large numbers of configuration policies – for example, tens of thousands of NFS exports. PlatformAPI calls can easily be scripted using 'curl' or 'wget' from UNIX/Linux, or via the Windows Powershell 'Invoke-RestMethod' cmdlet.

Some administration considerations include:

- WebUI limitations: Displaying large configuration output can be burdensome. Limited ability to sort and search output.
- The WebUI proves to be fairly responsive at scale, even in high load situations.
- CLI limitations: There are a number of OneFS CLI commands and tools which are node-local.
- For cluster-wide configuration and reporting with node-local commands, OneFS includes the 'isi_for_array' utility. Since isi_for_array executes serially across a cluster's nodes, it can take a considerable time to complete, or potentially time out, on large clusters.

① Be aware that platform API responses can become slow if a large cluster is heavily overloaded. Under these conditions, a syslog message along the following lines will be displayed: "OneFS API is temporarily unavailable. Try your request again." This can be mitigated by tuning the number of isi_papi_d child processes.

For example:

```
# isi_gconfig -t papi proc.max_children=80 && isi_for_array "killall HUP isi_papi_d"
```

Further information on the OneFS admin interfaces is available at:

 [OneFS Web Administration Guide](#)

 [OneFS CLI Administration Guide](#)

 [OneFS CLI Command Reference](#)

 [OneFS API Reference](#)

Cluster Upgrades

Upgrading a large cluster is a significant undertaking, likely involving down time and definitely requiring thorough planning and communication.

Non-disruptive upgrades (NDUs) allow a cluster administrator to upgrade the storage OS while their end users continue to access data without error or interruption. Updating the operating system on a OneFS powered cluster is a simple matter of a rolling upgrade. During this process, one node at a time is upgraded to the new code, and the active NFS and SMB3 Continuous Availability (CA) clients attached to it are automatically migrated to other nodes in the cluster. Partial upgrade is also permitted, whereby a subset of cluster nodes can be upgraded. The subset of nodes may also be grown during the upgrade.


① OneFS also supports upgrade rollback. This feature provides the ability to return a cluster with an uncommitted upgrade to its previous version of OneFS.

As part of an upgrade, OneFS automatically runs a pre-install verification check. This verifies that the configuration in your current installation of OneFS is compatible with the version of OneFS that is intended for the upgrade. When an unsupported configuration is found, the upgrade is stopped and instructions on troubleshooting the issue are displayed. Proactively running the pre-installation upgrade check before starting an upgrade helps to avoid any interruption due to incompatible configuration.

- For extra-large clusters, where possible plan on performing a full cluster reboot upgrade, since rolling upgrades can take days.
- OneFS 8.2.2 and later releases allow parallel upgrades, whereby clusters can upgrade an entire neighborhood, or fault domain, at a time, substantially reducing the duration of large cluster upgrades.
- Before running a major (non-rolling) OneFS upgrade, allow active jobs to complete, where possible, and cancel out any outstanding running jobs.
- Running the 'isi_flush' CLI utility to flush the data lock caches can help reduce maintenance and upgrade reboot times.
- Upgrade OneFS to a newer release at least once a year.
- On large clusters, careful planning and scheduling of firmware upgrades is required.

① At around 30 minutes per node, rolling firmware upgrades of extra-large clusters are undesirable most of the time due to the duration. Instead, most customers prefer to schedule a maintenance window and perform a full cluster reboot.

① An active MediaScan job that is running concurrently with a drive firmware update can significantly impact the performance on the cluster. To avoid this, pause any running MediaScan job before starting the firmware update process using the 'isi job pause mediascan' CLI command.

 Further information on updating OneFS and firmware is available in the [Upgrade Planning and Process guide](#).

Cluster Informational Resources and Dell EMC Support

Proactive cluster management is particularly important at large cluster scale. A significant part of this includes staying aware and informed about current issues, bugs, security advisories, etc. These are available at:

 [Dell EMC technical advisories \(ETAs\)](#)


 [Dell EMC security advisories](#)

① To sign up for regular product updates on the support site so you are automatically kept current on ETAs, KBs, new releases, etc:

1. Go to <https://support.emc.com> and click "Register here."
2. Follow the online registration steps. Make sure to fill in all required fields and use your business e-mail address.
3. Once your registration is processed, you will receive an e-mail confirming access and providing additional information to complete your registration and initial login.


Cluster Monitoring and Auditing

The cluster event logger (CELOG) is OneFS' alerting and reporting framework and is vital for keeping abreast of hardware and software issues across a large cluster.

 For a full listing and details of available CELOG events and alerts please reference the [OneFS Event Reference Guide](#).

For extra-large clusters, where group changes and other transient issues occur fairly frequently, the quantity of CELOG alerts and events management can become overwhelming. Alternate, and often preferable, approaches to cluster monitoring include:

- SNMP monitoring of a cluster's MIBs and traps from an external network management station (NMS).
- Syslog and cluster logfile monitoring.
- ESRS phone-home monitoring and support.

 A software development kit (SDK) is available for OneFS powered clusters. This SDK includes a Data Insights Connector which uses Grafana and InfluxDB to enable custom monitoring, alerts, and notifications. The SDK is freely available for download from [GitHub](#) and more information is available in the [installation instructions](#).

① A [OneFS App](#) is also available for Splunk Enterprise, which collects and reports on cluster configuration and performance data, plus some basic file system auditing. The app comes in two parts - a technology add-on (TA) that contains collection scripts, and the app itself which provides the visualizations.

InsightIQ

InsightIQ can be configured to monitor one or more large OneFS powered clusters. However, the recommendation is not to monitor a single cluster larger than 80 nodes and to monitor no more than 8 clusters or 150 nodes simultaneously with a single InsightIQ instance. As such, this is not an advised solution for extra-large clusters. For monitoring more than eight clusters, the best practice is to deploy an additional instance of InsightIQ.

 For more information on configuring and monitoring InsightIQ, please refer to the [Administration Guide](#).

EMC Secure Remote Services (ESRS)

OneFS communicates with the EMC backend using EMC Secure Remote Services (ESRS). ESRS is used to send alerts, log gathers, usage intelligence, and managed device status to the backend. Clusters provisioned with the EMC backend can download updates, patches, OneFS software packages, or any other file that has been designated for a cluster to download. ESRS provides data for technical support personnel to investigate and resolve cluster issues and support requests. In order to use this service, customers are required to host an ESRS Gateway, which securely proxies ESRS communication with the cluster.


① ESRS replaces the deprecated SupportIQ tool.

 For more information on deploying and configuring ESRS, please refer to the [Site Planning Guide](#).

Engaging Technical Support

If ESRS is not an option at your site, there are three ways to contact Dell EMC Technical Support directly:

- Live chat with a chat representative.
- Create a Service Request (SR) via the Dell EMC web portal.
- Contact EMC Technical Support by telephone.

 To get started, go to [Service Center](#) and log in with your credentials. On the right-hand pane, under the Service Center widget you will find links for the support engagement options.

① To ensure proper routing when initiating a Service Request (SR) ensure you reference the correct Site ID (Party Number) and Product Details.

Cluster Capacity Considerations

When a large cluster, or any of its nodepools, becomes more than 95% full, OneFS can experience slower performance and possible workflow interruptions in degraded mode and high-transaction or write-speed-critical operations. Furthermore, when a large cluster approaches full capacity (over 98% full), the following issues can occur:

- Performance degradation in some cases
- Workflow disruptions - failed file operations and inability to write data.
- Inability to make configuration changes or run commands to delete data and free up space
- Increased disk and node rebuild times.

To ensure that a cluster or its constituent pools do not run out of space:

- Add new nodes to existing clusters or pools
- Replace smaller-capacity nodes with larger-capacity nodes
- Create more clusters.

When deciding to add new nodes to an existing large cluster or pool, contact your sales team to order the nodes well in advance of the cluster or pool running short on space. The recommendation is to start planning for additional capacity when the cluster or pool reaches 75% full. This will allow sufficient time to receive and install the new hardware, while still maintaining sufficient free space.

The following table presents a suggested timeline for large cluster capacity planning:

Used Capacity	Action
75%	Plan additional node purchases.
80%	Receive delivery of the new hardware.
85%	Rack and install the new node(s).

Figure 11: Capacity Panning Timeline

Maintaining appropriate protection levels

Ensure your cluster and pools are protected at the appropriate level. Every time you add nodes, re-evaluate protection levels. OneFS includes a 'suggested protection' function that calculates a recommended protection level based on cluster configuration, and alerts you if the cluster falls below this suggested level

OneFS supports several protection schemes. These include the ubiquitous +2d:1n, which protects against two drive failures or one node failure, and +3d:1n1d, which guards against three drives or one node plus one drive failures.

① The best practice is to use the recommended protection level for a particular cluster configuration. This recommended level of protection is clearly marked as 'suggested' in the OneFS WebUI storage pools configuration pages and is typically configured by default. For all current Isilon Gen6 and PowerScale hardware configurations, the recommended protection levels are either '+2d:1n' or '+3d:1n1d'.

Monitoring cluster capacity

- **Configure alerts.** Set up event notification rules so that you will be notified when the cluster begins to reach capacity thresholds. Make sure to enter a current email address in order to receive the notifications.
- **Monitor alerts.** The cluster sends notifications when it has reached 95 percent and 99 percent capacity. On some clusters, 5 percent capacity remaining might mean that a lot of space is still available, so you might be inclined to ignore these notifications. On a large cluster, pay close attention to the alerts, closely monitor the cluster, and have a plan in place to take action when necessary.

- **Monitor ingest rate.** It is very important to understand the rate at which data is coming in to the cluster or pool.
- Options to do this include:
 - SNMP
 - SmartQuotas
 - FSAnalyze
- **Use SmartQuotas** to monitor and enforce administrator-defined storage limits. SmartQuotas manages storage use, monitors disk storage, and issues alerts when disk storage limits are exceeded. Although it does not provide the same detail of the file system that FSAnalyze does, SmartQuotas maintains a real-time view of space utilization so that you can quickly obtain the information you need.
- **Run FSAnalyze jobs.** FSAnalyze is a job-engine job that the system runs to create data for the InsightIQ file system analytics tools. FSAnalyze provides details about data properties and space usage within the /ifs directory. Unlike SmartQuotas, FSAnalyze updates its views only when the FSAnalyze job runs. Since FSAnalyze is a fairly low-priority job, it can sometimes be preempted by higher-priority jobs and therefore take a long time to gather all of the data. An InsightIQ license is required to run an FSAnalyze job.

Configuring Storage Quotas

One of the most useful of the OneFS data services for capacity allocation and management in large clusters is SmartQuotas. Quotas enable administrators to understand, predict, control and limit storage usage, and provision a cluster to best meet their storage needs. Furthermore, a default quota can be set on a top-level directory and be automatically inherited by subdirectories. This is ideal for automated home directory quota management across enterprises with large employee headcounts.

Quotas also facilitate ‘thin provisioning’, or the ability to present more storage capacity to applications and users than is physically present (over-provisioning). This allow for metered purchasing and provisioning of storage as the cluster grows, rather than having to make large, speculative purchasing decisions ahead of time. Here are a couple of examples of leveraging SmartQuotas in large cluster environments:

Scenario 1: Quota Management

A university has a large cluster which they use as a data lake. The IT department allocates their students and groups a fixed amount of capacity to control and keep storage growth in check. The storage administrator wants to know how much each student is consuming & limit them. To accomplish this, the storage admin:

- Sets default user hard or soft quotas
- Configures email alerts to students to encourage self-cleanup of file usage

Scenario 2: HPC Compute Farm Constraint

A semiconductor company uses a large HPC compute cluster for parts of their EDA workflow, and wants to guard against runaway jobs for consuming massive amounts of storage. The company runs heavy computations jobs from a large compute farm against a ‘scratch space’ directory, housed on a performance tier on their cluster, and garbage collection is run at midnight.

Throughout the workday, it’s hard for the storage admins to keep track of storage utilization. Occasionally, jobs from the compute farm run amok, tying up large swathes of fast, expensive storage resources and capacity. To help prevent this, the storage administrator:


- Sets an advisory directory quota on the scratch space at 80% utilization for advanced warning of an issue.
- Configures a hard directory quota to prevent writes at 90% utilization.

SmartQuotas best practices include:

- Leverage default quotas for ease of deployment and management at large scale. Configuration changes for linked quotas must be made on the parent quota that the linked quota is inheriting from. Changes to the parent quota are propagated to all children. To override configuration from the parent quota, you must unlink the quota first.

- Where possible, observe the best practice of a maximum number of 500,000 quotas per cluster in OneFS 8.2 and later, and 20,000 quotas per cluster in prior releases.
- Avoid creating quotas on the root directory of the default OneFS share (/ifs). A root-level quota may result in performance degradation.
- Limit quota depth to a maximum of 275 directories.
- Governing a single directory with overlapping quotas can also degrade performance.
- Directory quotas can also be used to alert of and constrain runaway jobs, preventing them from consuming massive amounts of storage space.
- The maximum tested quota limit is 400,000 (although the file system has no hard-coded limit on quotas). However, when listing a large number of quotas, only a partial list may be returned.
- With CloudPools data, the quota is calculated based on the size of the data local to the cluster. For example, for a 100MB file tiered to a cloud provider, SmartQuotas would calculate just the size of the local stub file (8K).
- If two quotas are created on the same directory – for example an accounting quota without Snapshots and a hard quota with Snapshots - the quota without Snapshot data overrides the limit from the quota with Snapshot data.
- SmartQuotas also provide a low impact way to provide directory file count reports.
- A quota can only be unlinked when it's linked to a default quota. Configuration changes for linked quotas must be made on the parent (default) quota that the linked quota is inheriting from. Changes to the parent quota are propagated to all children. If you want to override configuration from the parent quota, you must first unlink the quota.
- Quota containers compartmentalize /ifs, so that a directory with a container will appear as its own separate 'file system slice'. For example, to configure a directory quota with a 4TB container on /ifs/data/container1, you could use the following CLI command:

```
# isi quota quotas create /ifs/data/container1 directory --hard-threshold 4T --container true
```

 Further information is available in the [OneFS SmartQuotas](#) white paper.

In summary, best practices on planning and managing capacity on a large cluster include the following:

- To risk manage the possibility of adverse data delivery at very high capacity levels, Dell EMC recommends a maximum of 85% capacity utilization (also stated as a Reserve Capacity of 15%) on large clusters.
- At any workload level, do not exceed 90% capacity utilization.
- Examine data delivery variance at 75% and again at 80% capacity utilization; evaluate consistency of data delivery before using additional capacity.
- Consider a buffer for maintenance actions (disk rebuild) when planning reserve capacity.
- Maintain sufficient free space.
- Plan for contingencies.
- Manage your data.
- Maintain appropriate protection levels.
- Monitor cluster capacity and data ingest rate.
- Consider configuring a separate accounting quota for '/ifs/home' and '/ifs/data directories' (or wherever data and home directories are provisioned) to monitor aggregate disk space usage and issue administrative alerts as necessary to avoid running low on overall capacity.
- Ensure that any snapshot, replication and backup schedules meet the required availability and recovery objectives and fit within the overall capacity.
- Carefully manage snapshot creation and deletion schedules.
- Leverage SmartQuotas to understand, predict, control and limit storage usage.
- Use InsightIQ, CloudIQ, and DataIQ for monitoring, usage forecasting, and verifying cluster health.

OneFS Software Considerations

Cluster Composition and Group State

One of the most significant impacts to a large cluster's workflow at large scale is the effect of group changes resulting from the addition, removal, or rebooting of a node, or other hardware failure or transience. Having the ability to understand a cluster's group state and changes is an invaluable tool when administering and managing large clusters. It allows you to determine the current health of a cluster, as well as reconstruct the cluster's history when troubleshooting issues that involve cluster stability, network health, etc.

The primary role of the OneFS Group Management Protocol (GMP) is to help create and maintain a group of synchronized nodes. A group is a given set of nodes which have synchronized state, and a cluster may form multiple groups as connection state changes. GMP distributes a variety of state information about nodes and drives, from identifiers to usage statistics. The most fundamental of these is the composition of the cluster, or 'static aspect' of the group, which is managed by the `isi_boot_d` daemon and stored in the `array.xml` file.

Similarly, the state of a node's drives is stored in the `drives.xml` file, along with a flag indicating whether the drive is an SSD. Whereas GMP manages node states directly, drive states are actually managed by the 'drv' module, and broadcast via GMP. A significant difference between nodes and drives is that for nodes, the static aspect is distributed to every node in the `array.xml` file, whereas drive state is only stored locally on a node.

A group change operation, based on GMP, is a coherent way of changing the cluster-wide shared state. Merge is the group change operation for addition of nodes. Merges affect cluster availability due to their need to pause any filesystem operations for the duration of the operation. The `array.xml` information is needed by every node in order to define the cluster and allow nodes to form connections. In contrast, `drives.xml` is only stored locally on a node. If a node goes down, other nodes have no method to obtain the drive configuration of that node. Drive information may be cached by the GMP, but it is not available if that cache is cleared.

Conversely, 'dynamic aspect' refers to the state of nodes and drives which may change. These states indicate the health of nodes and their drives to the various file system modules - plus whether or not components can be used for particular operations. For example, a soft-failed node or drive should not be used for new allocations. These components can be in one of seven states:

Component State	Description
UP	Component is responding
DOWN	Component is not responding
DEAD	Component is not allowed to come back to the UP state and should be removed.
STALLED	Drive is responding slowly.
GONE	Component has been removed.
Soft-failed	Component is in the process of being removed.
Read-only	This state only applies to nodes.

Figure 12: OneFS Group Management – Component States

① A node or drive may go from 'down, soft-failed' to 'up, soft-failed' and back. These flags are persistently stored in the `array.xml` file for nodes and the `drives.xml` file for drives.

Group and drive state information allows the various file system modules to make timely and accurate decisions about how they should utilize nodes and drives. For example, when reading a block, the selected mirror should be on a node and drive where a read can succeed (if possible). File system modules use the GMP to test for node and drive capabilities, which include:

Node Capability	Description
Readable	Drives on this node may be read.
Writable	Drives on this node may be written to.
Restripe From	Move blocks away from the node.

Figure 13: OneFS Group Management – Node Capabilities

Access levels help define 'as a last resort' with states for which access should be avoided unless necessary. The access levels, in order of increased access, are as follows:

Access Level	Description
Normal	The default access level.
Read Stalled	Allows reading from stalled drives.
Modify Stalled	Allows writing to stalled drives.
Read Soft-fail	Allows reading from soft-failed nodes and drives.
Never	Indicates a group state never supports the capability.

Figure 14: OneFS Group Management – Access Level

Drive state and node state capabilities are shown in the following tables. As shown, the only group states affected by increasing access levels are soft-failed and stalled.

Minimum Access Level for Capabilities Per Node State

Node States	Readable	Writable	Restripe From
UP	Normal	Normal	No
UP, Smartfail	Soft-fail	Never	Yes
UP, Read-only	Normal	Never	No
UP, Smartfail, Read-only	Soft-fail	Never	Yes
DOWN	Never	Never	No
DOWN, Smartfail	Never	Never	Yes
DOWN, Read-only	Never	Never	No
DOWN, Smartfail, Read-only	Never	Never	Yes
DEAD	Never	Never	Yes

Figure 15: OneFS Group Management - Node State Capabilities

Minimum Access Level for Capabilities Per Drive State

Drive States	Minimum Access Level to Read	Minimum Access Level to Write	Restripe From
UP	Normal	Normal	No

UP, Smartfail	Soft-fail	Never	Yes
DOWN	Never	Never	No
DOWN, Smartfail	Never	Never	Yes
DEAD	Never	Never	Yes
STALLED	Read_Stalled	Modify_Stalled	No

Figure 16: OneFS Group Management - Drive State Capabilities

OneFS depends on a consistent view of a cluster's group state. For example, some decisions, such as choosing lock coordinators, are made assuming all nodes have the same coherent notion of the cluster.

Group changes originate from multiple sources, depending on the particular state. Drive group changes are initiated by the drv module. Service group changes are initiated by processes opening and closing service devices. Each group change creates a new group ID, comprising a node ID and a group serial number. This group ID can be used to quickly determine whether a cluster's group has changed, and is invaluable for troubleshooting cluster issues, by identifying the history of group changes across the nodes' log files.

GMP provides coherent cluster state transitions using a process similar to two-phase commit, with the up and down states for nodes being directly managed by the GMP. The Remote Block Manager (RBM) provides the communication channel that connect devices in the OneFS. When a node mounts /ifs, it initializes the RBM in order to connect to the other nodes in the cluster and uses it to exchange GMP information, negotiate locks, and access data on the other nodes.

Before /ifs is mounted, a 'cluster' is just a list of MAC and IP addresses in array.xml, managed by isi_boot_d when nodes join or leave the cluster. When mount_efs is called, it must first determine what it's contributing to the file system, based on the information in drives.xml. After a cluster (re)boot, the first node to mount /ifs is immediately placed into a group on its own, with all other nodes marked down. As the Remote Block Manager (RBM) forms connections, the GMP merges the connected nodes, enlarging the group until the full cluster is represented. Group transactions where nodes transition to UP are called a 'merge', whereas a node transitioning to down is called a split. Several file system modules must update internal state to accommodate splits and merges of nodes. Primarily, this is related to synchronizing memory state between nodes.

The soft-failed, read-only, and dead states are not directly managed by the GMP. These states are persistent and must be written to array.xml accordingly. Soft-failed state changes are often initiated from the user interface, for example via the 'isi devices' command.

A GMP group relies on cluster quorum to enforce consistency across node disconnects. Requiring $[N/2]+1$ replicas to be available ensures that no updates are lost. Since nodes and drives in OneFS may be readable, but not writable, OneFS has two quorum properties:

- Read quorum
- Write quorum

Read quorum is governed by having $[N/2] + 1$ nodes readable, as indicated by `sysctl efs.gmp.has_quorum`. Similarly, write quorum requires at least $[N/2] + 1$ writeable nodes, as represented by the `sysctl efs.gmp.has_super_block_quorum`. A group of nodes with quorum is called the 'majority' side, whereas a group without quorum is a 'minority'. By definition, there can only be one 'majority' group, but there may be multiple 'minority' groups. A group which has any components in any state other than up is referred to as degraded.

File system operations typically query a GMP group several times before completing. A group may change over the course of an operation, but the operation needs a consistent view. This is provided by the group info, and includes the GMP's group state, plus information about services provided by nodes in the cluster. This allows nodes in the cluster to discover when services go up or down on other nodes and take the appropriate action when that occurs.

Processes also change the service state in GMP by opening and closing service devices. A particular service transitions from down to up in the GMP group when it opens the file descriptor for a service-specific device. Closing the service file descriptor will trigger a group change that reports the service as down. A process can explicitly close the file descriptor if it chooses, but most often the file descriptor will remain open for the duration of the process and closed automatically by the kernel when it terminates.

Understanding and Analyzing Group Membership

Group membership is one of the key troubleshooting tools for large clusters, where the group composition may change fairly frequently as drives and other components degrade and are SmartFailed out. As such, an understanding of OneFS group membership reporting allows you to determine the current health of a cluster. It also enables a cluster's history to be reconstructed when triaging and troubleshooting issues that involve cluster stability, network health, and data integrity.

Under normal operating conditions, every node and its requisite disks are part of the current group. This can be viewed by running the 'sysctl efs.gmp.group' CLI command from any healthy node of the cluster. A OneFS group is comprised of two parts:

- Sequence number: Provides identification for the group.
- Membership list: Describes the group.

The membership list shows the group members within brackets. Consider the following example:

```
{ 1-3:0-11 }
```

This represents a healthy three node X210 cluster, with Node IDs 1 through 3. Each node contains 12 hard drives, numbered zero through 11.

- The numbers before the colon in the group membership list represent the participating Array IDs.
- The numbers after the colon represent Drive IDs.

Array IDs differ from Logical Node Numbers (LNNs), the node numbers that occur within node names, and displayed by `isi stat`. These numbers may also be retrieved on a node via the `isi_nodes` command. LNNs may be re-used, whereas Array IDs are never re-used. Drive IDs are also never recycled. When a drive is failed, the cluster will identify the replacement drive with the next unused number. However, unlike Array IDs, Drive IDs start at 0 rather than at 1.

Group messages also compact sequential lists into a pair of numbers separated by dashes, as in our previous example of { 1-3:0-11 }. Without dashes, deployed to compress the node list, this group might be listed as:

```
{ 1:0-11, 2:0-11, 3:0-11 }
```

When drive 2 is removed from node 2, this consolidated list would become:

```
{ 1:0-11, 2:0-1,3-11, 3:0-11 }
```

When a replacement disk is added to node 2, the list would become:

```
{ 1:0-11, 2:0-1,3-12, 3:0-11 }.
```

Over time and as the node count grows, these changes can make cluster groups considerably more challenging to read.

Identifying Accelerator Nodes

Accelerator nodes appear differently in group messages, because they have no disks to be part of the group. In fact, accelerators ("diskless" nodes) appear twice, once as a node with no disks, and again explicitly as a diskless node.

For example, consider the group:

```
{ 1:0-23, 2,4:0-10,12-24, 5:0-10,12-16,18-25, 6:0-17,19-24, 7:0-10,12-24, 9-10:0-23, 11:0-3,5-24, 12-13, 14:0-23, diskless: 12-13 }
```

Nodes 12 and 13 are listed both as diskless, but also listed between nodes 11 and 14, albeit with no drives.

SmartFailed and down nodes

Similar to accelerators, nodes in a SmartFail state show up both separately and in the regular group membership. For example, in the group:

```
{ 1-4:0-23, soft_failed: 1 }
```

Node 1 has been SmartFailed, but it is still part of the group. When the FlexProtect completes, the node will be removed from the group.

When a node has been SmartFailed, but is also unavailable, it will be listed as `soft_failed` but not listed as part of the group. For example:


```
{ 1-4:0-23, 6:0-17,19-24, down: 5, soft_failed: 5 }
```

Shows node 5 as both down and SmartFailed.

When a node is offline, other nodes will show that node as down, as in:

```
{ 3-4:0-8, 5:0-6,8, 9:1-2,4-6,8, 12:0-11, down: 6 }
```

Note that no disks for that node are listed, and that it doesn't show up in the group.

If the node is split from the cluster—that is, if it is online but not able to contact other nodes on its back-end network—that node will see the rest of the cluster as down. Its group might look something like { 6:0-11, down: 3-5,8-9,12 } instead.

When calculating whether a cluster is below protection level, SmartFailed devices should be considered 'in the group' unless they are also down: a cluster with +2:1 protection with three nodes up but smartfailed does not pose an exceptional risk to data availability.

SmartFailed and down drives

Like nodes, drives may be smartfailed and down, or smartfailed but available. The group statement looks similar to that for a smartfailed or down node, only the drive Lnum is also included. For example:

```
{ 1-4:0-23, 5:0-6,8-23, 6:0-17,19-24, down: 5:7, soft_failed: 5:7 }
```

This indicates that node ID 5 drive Lnum 7 is both SmartFailed and unavailable.

If the drive was SmartFailed but still available, the group would read:

```
{ 1-4:0-23, 5:0-6,8-23, 6:0-17,19-24, soft_failed: 5:7 }.
```

When multiple devices are down, consolidated group statements can be tricky to read. For example, if node 1 was down, and drive 4 of node 3 was down, the group statement would read:

```
{ 2:0-11, 3:0-3,5-11, 4-5:0-11, down: 1, 3:4, soft_failed: 1, 3:4 }
```

However, if node 1 was up but drive 4 on node 1 was down, the group statement would read:

```
{ 1:0-3,5-11, 2:0-11, 3:0-3,5-11, 4-5:0-11, down: 1,3:4, soft_failed: 1,3:4 }
```

Read-only nodes

A node that has been placed in read-only mode can be clearly identified in the group:

```
{ 1-6:0-8, soft_failed: 2, read_only: 3 }.
```

Node 3 is shown both as a regular group member and called out separately at the end, like accelerators and nodes that have been SmartFailed but are nevertheless still active.

Also note that the term "read only" indicates that OneFS will not write to the disks in that node; incoming connections to the node can still write to other nodes of the cluster.

Dead nodes

Dead nodes appear in groups when a node has been permanently removed from the cluster without SmartFailing the node. These appear similar to down nodes, except that they are marked as dead rather than down:

```
{ 1-5:0-11, 6:0-7,9-12, 7-10,12-14:0-11, 15:0-10,12, 16-17:0-11, dead: 11 }
```

① If confronted with a dead node, the best course of action is to immediately contact Dell EMC Support and immediately start a FlexProtect job.

Dead drives

Drives in a dead state look similar to dead nodes, only they include a drive number as well as a node number. For example:

```
{ 1:0-11, 2:0-9,11, 3:0-11, 4:0-11, 5:0-11, 6:0-11, dead: 2:10 }
```

Dead drives can occur when a disk is simply unresponsive to any level of request from the operating system, or when a drive is removed from the node and replaced without starting a FlexProtect. On encountering a dead disk, contact Dell EMC Support and execute FlexProtect via the Job Engine to stripe away from that disk.

SmartFailed and stalled drives

SmartFailed disks appear in a similar fashion to other drive-specific states, and therefore include both an array ID and a drive ID. For example:

```
{ 1:0-11, 2:0-3,5-12, 3-4:0-11, 5:0-1,3-11, 6:0-11, soft_failed: 5:2 }
```

This shows drive 2 in node 5 to be SmartFailed but still available. If the drive was physically unavailable or damaged beyond communication with the node, the group would be presented as:

```
{ 1:0-11, 2:0-3,5-12, 3-4:0-11, 5:0-1,3-11, 6:0-11, down: 5:2, soft_failed: 5:2 }
```

Stalled drives (drives that don't respond) are shown similarly to down drives, for example:

```
{ 1:0-2,4-11, 2-4:0-11, stalled: 1:3 }
```

When a drive is 'un-stalled', it simply returns to the group.

① A large number of stalled drive messages may indicate a performance issue.

Reading Group Sequence Numbers

A group displays the sequence number between angle brackets. For example, <3,6>: { 1-3:0-11 }, the sequence number is <3,6>.

The first number within the sequence, in this case 3, identifies the node that initiated the most recent group change.

In the case of a node leaving the group, the lowest-numbered node remaining in the cluster will initiate the group change and thus appear as the first number within the angle brackets. In the case of a node joining the group, the newly-joined node will initiate the change and thus will be the listed node. If the group change involved a single drive joining or leaving the group, the node containing that drive will initiate the change and thus will be the listed node. The second piece of the group sequence is the counter, which increases sequentially.

Group Changes

Group changes may be caused by drive removals or replacements, node additions, node removals, node reboots or shutdowns, backend (internal) network events, and the transition of a node into read-only mode. For debugging purposes, group change messages can be reviewed to determine whether any devices are currently in a failure state.

When a group change occurs, a cluster-wide process writes a message describing the new group membership to /var/log/messages on every node. Similarly, if a cluster 'splits', the newly-formed clusters behave in the same way: Each node records its group membership to /var/log/messages. When a cluster splits, it breaks into multiple clusters (multiple groups). Rarely, if ever, is this a desirable event.

① The terms 'cluster' and 'group' are synonymous: A cluster is defined by its group members. Group members which lose sight of other group members no longer belong to the same group and thus no longer belong to the same cluster.

To view group changes from an individual node's perspective, 'grep' for the expression 'new group' to extract the group change statements from that node's 'messages' logfile. For example:

```
X210-1# grep -i 'new group' /var/log/messages | tail -n 10
```

```
May 30 08:07:50 (id1) /boot/kernel/kernel: [gmp_info.c:530] (pid 1814="kt: gmpdrive-upda") new group: : { 1:0-4, down: 1:5-11, 2-3 }
```

```
May 30 08:07:50 (id1) /boot/kernel/kernel: [gmp_info.c:530] (pid 1814="kt: gmpdrive-upda") new group: : { 1:0-5, down: 1:6-11, 2-3 }
```

```
May 30 08:07:50 (id1) /boot/kernel/kernel: [gmp_info.c:530] (pid 1814="kt: gmpdrive-upda") new group: : { 1:0-6, down: 1:7-11, 2-3 }
```

```

May 30 08:07:50 (id1) /boot/kernel/kernel: [gmp_info.c:530] (pid 1814="kt: gmpdrive-upda") new
group: : { 1:0-7, down: 1:8-11, 2-3 }

May 30 08:07:50 (id1) /boot/kernel/kernel: [gmp_info.c:530] (pid 1814="kt: gmpdrive-upda") new
group: : { 1:0-8, down: 1:9-11, 2-3 }

May 30 08:07:50 (id1) /boot/kernel/kernel: [gmp_info.c:530] (pid 1814="kt: gmpdrive-upda") new
group: : { 1:0-9, down: 1:10-11, 2-3 }

May 30 08:07:50 (id1) /boot/kernel/kernel: [gmp_info.c:530] (pid 1814="kt: gmpdrive-upda") new
group: : { 1:0-10, down: 1:11, 2-3 }

May 30 08:07:50 (id1) /boot/kernel/kernel: [gmp_info.c:530] (pid 1814="kt: gmpdrive-upda") new
group: : { 1:0-11, down: 2-3 }

May 30 08:07:51 (id1) /boot/kernel/kernel: [gmp_info.c:530] (pid 1814="kt: gmpmerge") new group: : {
1:0-11, 3:0-7,9-12, down: 2 }

May 30 08:07:52 (id1) /boot/kernel/kernel: [gmp_info.c:530] (pid 1814="kt: gmpmerge") new group: : {

```

① In this case, the tail -10 command has been used to limit the returned group change output to the last ten reported in the file. All of these occur within two seconds, so you would want to go further back, to before whatever incident was under investigation.

Interpreting Group Changes

Even in the example above we can be certain of several things:

- At last report all nodes of the cluster are operational and joined into the cluster. No nodes or drives report as down or split. (At some point in the past, drive ID 8 on node 3 was replaced, but a replacement disk has been added successfully.)
- Node 1 rebooted: for the first eight out of ten lines, each group change is adding back a drive on node 1 into the group, and nodes two and three are inaccessible. This occurs on node reboot prior to any attempt to join an active group and is correct and healthy behavior.
- Node 3 joins in with node 1 before node 2 does. This might be coincidental, given that the two nodes join within a second of each other. On the other hand, perhaps node 2 also rebooted while node 3 remained up. A review of group changes from these other nodes could confirm either of those behaviors.

In this case, a check of node 2 and 3's logs will confirm whether they also rebooted at the same time indicating a cluster-wide event.

Constructing an event timeline

When investigating a large cluster issue, it can be helpful to build a human-readable timeline of what occurred. This is particularly useful in instances with multiple, non-simultaneous group changes. This timeline should include which nodes have come up or down and can be interpolated with panic stack summaries to describe an event.

Extra-large cluster example

With large clusters, the group state output can often be complex and difficult to parse. For example, consider the following group state report from an extra-large cluster:

```

# sysctl efs.gmp.group

efs.gmp.group: <47,3501>: { 1:0-22, 2-3:0-17, 4, 5:0-11, 6-10:0-22, 11-13:0-23, 14:0-11,13-19,21-
23, 15:0-22, 16-31:0-23, 32-38:0-17, 39-41:0-35, 42:0-14,16-35, 43-45:0-33, 46-48:0-35, 49-53:0-22,
54-69, 70-80:0-11, 81:0-10, 82-89,91-93,95-126:0-11, 127-129:0-10, 130-133:0-11, diskless: 4, 54-
69, smb: 1-89,91-93,95-133, nfs: 1-89,91-93,95-133, hdfs: 1-89,91-93,95-133, all_enabled_protocols:
1-89,91-93,95-133, isi_cbind_d: 1-89,91-93,95-133, lsass: 1-89,91-93,95-133, s3: 1-89,91-93,95-133
}

```

From this output, we can make determinations such as:

- The cluster comprises 131 nodes, with ID's 90 and 94 unused (1-89,91-93,95-133)
- NFS, HDFS, SMB and S3 protocols are running on all nodes (all_enabled_protocols: 1-89,91-93,95-133)

- A100 Diskless accelerators comprise 17 of the cluster's nodes (4, 54-69)
- Node 1 has 23 drives so is an Isilon S-series node, either with 1 SSD or a failed drive (1:0-22)
- Nodes 2 and 3 are Isilon X2*0 nodes, each with 18 disks (2-3:0-17)
- Nodes 39-14 have 36 drives each, so are 4RU NL nodes without SSDs
- Node 14 has drives 12 and 20 missing (14:0-11,13-19,21-23)

① If more detail is desired, the `sysctl efs.gmp.current_info` command will report extensive current GMP info.

Additional OneFS group management considerations include:

- Having a number of drive outages/failures can cause considerable group state churn. As such, the best practice is to promptly replace any failed drives.
- Within OneFS, the `/ifs/.ifsvar` directory contains most of the cluster's configuration and contextual information. With a high node count and in heavily degraded conditions, GMP can still have quorum with eight nodes down. In such a situation, there may be portions of the `/ifs/.ifsvar` directory structure that are unavailable.
- Assuming all nodes are connected to the network, be aware of the impact of group changes. For instance, if a node is rebooted, the back-end updates between nodes can be disruptive for some protocols and applications.
- Other protocols like NFS and SMB3 continuous availability will gracefully handle the disruption.
- Avoid direct IP connections (non-dynamic) to the cluster by using the SmartConnect VIP.
- At large cluster scale, a group change resulting from adding/removing/rebooting a node, etc, can impact I/O for 15 seconds or more. Similarly, a drive stall event can delay an I/O for 2 or more seconds.
- The following sysctls can help reduce excessive GMP activity on a busy extra-large cluster by increasing its tolerance to ping timeouts. Only modify these settings under the direction and supervision of Dell EMC Support.

```
# isi_sysctl_cluster efs.rbm.dwt_handle_pings=1
# isi_sysctl_cluster net.inet.sdp.fin_wait_timeout=10
# isi_sysctl_cluster net.inet.sdp.time_wait_timeout=3
```

- In OneFS, there is no automatic repair job started when a node is lost. It requires manual intervention. In the past, OneFS did have a concept of 'down for time' timeout after which FlexProtect would start in the presence of a down node. This didn't work well in practice given the transient nature of some node failures (plus maintenance, etc), and ended up causing more repair work to get done (initial repair, plus the 'un-repair' when the node was returned to the group).
- With newer nodes having swappable journals and with disk tango a more frequent function, fixing a node and returning it to service is more commonplace nowadays.
- In OneFS 8.0 and later, the SmartConnect process will continue to give out IP addresses during a group merge or split.
- OneFS 8.2 and later sees `isi_boot_d` replaced by `isi_array_d` and the adoption of the Paxos protocol as part of the infrastructure scale enhancements to support 252 node clusters.
- OneFS 9.0 sees the addition of the S3 object protocol.

 Further information is available in the [OneFS Cluster Composition, Quorum, and Group State](#) white paper.

Layout, Protection, and Failure Domains

OneFS provisioning works on the premise of dividing similar nodes' drives into sets, or disk pools, with each pool representing a separate failure domain. These are protected by default at +2d:1n (or the ability to withstand two drive or one entire node failure), or often +3d:1n1d (three drive or one node and one drive failure) in larger and denser clusters.

Unlike the PowerScale F600 and F200 nodes and previous Isilon hardware generations, where a single node was self-contained, each Isilon Gen6 platform chassis contains four compute modules (one per node), and five drive containers, or sleds, per node. Each sled is a tray which slides into the front of the chassis and contains between three and six drives, depending on the configuration of a particular chassis.

Multiple groups of different node types, or node pools, can work together in a single, heterogeneous cluster. For example: One node pool of F-series nodes for I/Os-intensive applications, one node pool of H-series nodes, primarily used for high-concurrent and sequential workloads, and one node pool of A-series nodes, primarily used for nearline and/or deep archive workloads.

This allows a large cluster to present a single storage resource pool comprising multiple drive media types – SSD, high speed SAS, large capacity SATA, etc - providing a range of different performance, protection and capacity characteristics. This heterogeneous storage pool in turn can support a diverse range of applications and workload requirements with a single, unified point of management. It also facilitates the mixing of older and newer hardware, allowing for simple investment protection even across product generations, and seamless hardware refreshes.

Each Node Pool only contains disk pools from the same type of storage nodes and a disk pool may belong to exactly one node pool. Any new node added to a cluster is automatically allocated to a node pool and then subdivided into disk pools without any additional configuration steps, inheriting the SmartPools configuration properties of that node pool

Tiers

Tiers are groups of node pools combined into a logical superset to optimize data storage, according to OneFS platform type. For example, similar 'archive' node pools are often consolidated into a single tier. This tier could incorporate different styles of Isilon H-series and A-series node pools into a single, logical container. This is a significant benefit because it allows customers who consistently purchase the highest capacity nodes available to consolidate a variety of node styles within a single group, or tier, and manage them as one logical group.

- ① The recommendation for large clusters is to configure no more than two tiers of storage.

Storage pools

<div> <div>Summary</div> <div>File pool policies</div> <div>SmartPools</div> <div>CloudPools</div> <div>SmartPools settings</div> <div>CloudPools settings</div> </div>							
Tiers and pools							Create a tier
Type/Name	State	Nodes	Protection Level	L3 cache	Capacity	Actions	
- Tier: Tier1	--	1-19, ...	--	N/A	HDD: 0.13% used, 99.87% available	✕	
Pool: h500_30tb_3.2tb-ssd_128gb	--	1-19, ...	+2d:1n	Enabled	HDD: 0.12% used, 99.88% available	✕	
Pool: h600_18tb_3.2tb-ssd_256gb	--	37, 52,...	+2d:1n	Enabled	HDD: 0.17% used, 99.83% available	✕	
- Tier: Tier2	--	20-36,...	--	N/A	HDD: 0.22% used, 99.78% available	✕	
Pool: a2000_200tb_800gb-ssd_16c	--	20-36,...	+2d:1n	Enabled	HDD: 0.23% used, 99.77% available	✕	
Pool: a200_30tb_800gb-ssd_16gb	--	40-51	+2d:1n	Enabled	HDD: 0.14% used, 99.86% available	✕	

Figure 17: SmartPools WebUI View – Tiers and Node Pools

Failure Domains & Neighborhoods

The Isilon Gen6 platform, where four nodes are contained in a 4RU chassis, enhances the concept of disk pools, node pools, and 'neighborhoods' to add another level of resilience into the OneFS failure domain concept.

The Isilon Gen6 chassis architecture provides three fundamental areas of hardware resilience, in addition to data protection via OneFS erasure coding. These are of particular importance in large clusters, where the sheer scale of componentry involved suggests a higher rate of hardware failure. These three areas of resilience are:

- Drive sled protection
- Partner node protection
- Chassis protection

In OneFS, a failure domain is the portion of a dataset that can be negatively impacted by a specific component failure. A disk pool comprises a group of drives spread across multiple compatible nodes, and a node usually has drives in multiple disk pools which share the same node boundaries. Since each piece of data or metadata is fully contained within a single disk pool, OneFS considers the disk pool as its failure domain.

With sled protection, each drive in a sled is automatically located in a different disk pool. This ensures that if a sled is removed, rather than a failure domain losing four drives, the affected failure domains each only lose one drive.

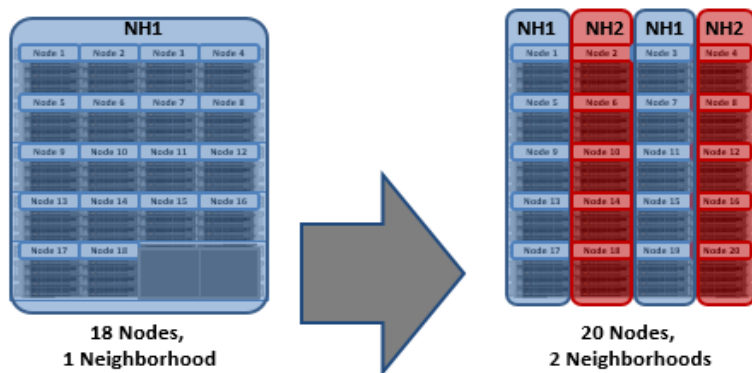
For larger clusters, 'neighborhoods' help organize and limit the width of a disk pool. Neighborhoods also contain all the disk pools within a certain node boundary, aligned with the disk pools' node boundaries. As such, a node will often have drives in multiple disk pools, but a node will only be in a single neighborhood.

① Neighborhoods, node pools, and tiers are all layers on top of disk pools. Node pools and tiers are used for organizing neighborhoods and disk pools.

The primary purpose of neighborhoods is to improve reliability of large clusters, and guard against data unavailability from the accidental removal of drive sleds, for example. For self-contained nodes like the PowerScale F600, OneFS has an ideal size of 20 nodes per node pool, and a maximum size of 39 nodes. On the addition of the 40th node, the nodes split into two neighborhoods of twenty nodes. However, for the Isilon Gen6 modular chassis-based platform, the ideal size of a neighborhood changes from 20 to 10 nodes. This protects against simultaneous node-pair journal failures and full chassis failures.

Partner nodes are nodes whose journals are mirrored. With the Isilon Gen6 platform, rather than each node storing its journal in NVRAM as in previous platforms, the nodes' journals are stored on SSDs - and every journal has a mirror copy on another node. The node that contains the mirrored journal is referred to as the partner node. There are several reliability benefits gained from the changes to the journal. For example, SSDs are more persistent and reliable than NVRAM, which requires a charged battery to retain state. Also, with the mirrored journal, both journal drives have to die before a journal is considered lost. As such, unless both of the mirrored journal drives fail, both of the partner nodes can function as normal.

With partner node protection, where possible, nodes will be placed in different neighborhoods - and hence different failure domains. Partner node protection is possible once the cluster reaches five full chassis (20 nodes) when, after the first neighborhood split, OneFS



places partner nodes in different neighborhoods:

Partner node protection increases reliability because if both nodes go down, they are in different failure domains, so their failure domains only suffer the loss of a single node.

For larger clusters, chassis protection ensures that each of the four nodes within each chassis will be placed in a separate neighborhood. Chassis protection becomes possible at 40 nodes, as the neighborhood split at 40 nodes enables every node in a chassis to be placed in a different neighborhood. As such, when a 38-node Isilon Gen6 cluster is expanded to 40 nodes, the two existing neighborhoods will be split into four 10-node neighborhoods:

Figure 18: OneFS Neighborhood Split – 20 Nodes

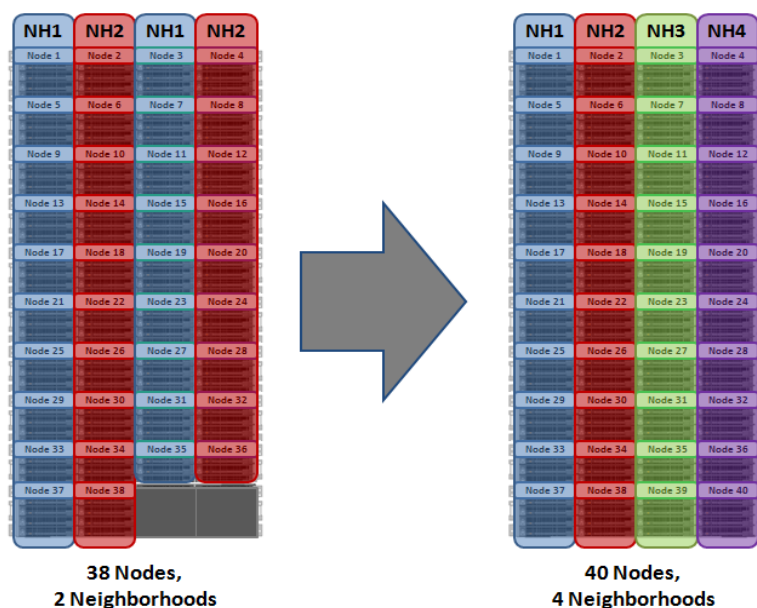


Figure 19: OneFS Neighborhood Split – 40 Nodes

Chassis protection ensures that if an entire chassis failed, each failure domain would only lose one node.

So, in summary, an Isilon Gen6 platform cluster running will have a reliability of at least one order of magnitude greater than previous generation clusters of a similar capacity as a direct result of the following enhancements:

- Mirrored Journals
- Neighborhoods
- Mirrored Boot Drives

Note that during the neighborhood split process, the job engine will need to run to complete the needed REPROTECT as you add new nodes. There may be up to a 20% performance impact to writes and some read operations during this time. The load mitigation options available are either to reduce the impact level of the job to 'LOW' or to disable the job entirely during peak times. However, clearly the downside to both these options is an increase in the time to job completion.

File Layout and Directory Structure

In general, it is more efficient to create a deep directory hierarchy that consolidates files in balanced subdirectories than it is to spread files out over a shallow subdirectory structure. This is particularly true for large clusters.

Although the recommended maximum file limit per directory is one million, a best practice is to constrain the number of files in any one directory to one hundred thousand. A maximum of 100,000 directories per directory is also recommended.

The general rules of thumb are to try to:

1. Keep the number of entries in a directory to a reasonable value.
2. Avoid storing unrelated data in the same directory.

Since OneFS is a hierarchical filesystem, these will help prevent creation of a data hot spot. Since directory updates (creates and deletes) are serialized, only one can happen at once. As such, a create/delete-heavy workload can benefit from as much as two orders of magnitude better performance when the data is organized into a hundred directories each containing ten thousand files compared to a single directory with a million files.


① The key for file and directory layout always revolves around balance. The recommended goal is for a directory tree structure and its file contents is to be as uniform as possible.

- Storing large numbers of files in a directory may affect enumeration and performance, but whether performance is affected depends on workload, workflow, applications, tolerance for latency, and other factors. To better handle storing a large number of files in a directory, use nodes that contain solid state drives (SSDs).
- Directory tree depth is limited to 509 directories and is determined by a maximum path length of 1,023 characters. However, depths greater than 275 directories may affect system performance.
- The maximum number of open files is 315,000 per node.
- Hard links are limited to a maximum of 65,535 per file. However, setting the number of per-file hard links to higher than 1,000 can slow down snapshot operations and file deletions. This per-file value can be configured via the sysctl 'efs.ifm.max_links'.
- The OneFS protocol daemons, such as the input-output daemon (lwio), may impose additional constraints on the number of files that a node can have open. The protocol daemons typically impose such constraints because the kernel places limits on per-process memory consumption.
- The largest file size that OneFS currently supports is increased to 16TB in OneFS 8.2.2, up from a maximum of 4TB in prior releases.

A typical data set consists of a mix of large and small files stored in a file system comprising a hierarchical directory structure. Usually, around 30 percent of the data is active; 70 percent is inactive. Snapshots typically back up the data for short-term retention combined with a long-term DR strategy, which frequently includes replication to a secondary cluster, and disk-to-disk or disk to tape NDMP backups.

OneFS uses erasure coding (FEC) to parity protect a file, which results in high levels of storage efficiency. Conversely, files less than 128KB in size are essentially mirrored, so have a larger on-disk footprint. Large file efficiency via erasure coding offsets the penalty of mirroring of small files.

While OneFS is highly extensible and a large number of the configuration options are physically unbounded, there are a number of recommended limits that do apply – especially in the context of large clusters. For example, the recommendation is to configure a maximum of forty thousand NFS exports on a single cluster. Even though it is possible to create more than this, the best practice is bounded at no more than forty thousand. Often, these 'limit' numbers are the maximum that Dell EMC engineering have tested and certified a cluster to. While things may work fine beyond these thresholds, such configurations will typically be unsupported.

 Further information on OneFS limits and guidelines is available in the [OneFS Technical Specifications](#) guide.

Data Layout and Tiering Recommendations

OneFS SmartPools enables large multi-tier clusters to be created using high performance nodes with SSD for performance tiers and high-capacity SATA-only nodes for the high-capacity archive tier. For example, a file pool policy could move files from the performance tier to a more cost-effective capacity-biased tier after the desired period of inactivity.

When using SmartPools tiering with large heterogeneous clusters, there are some cardinal rules of thumb to be cognizant of:

File pool rules needs to be carefully crafted:

- Simpler is always better
- Avoid storing snapshots on a slower tier than the head data.
- Be especially careful with any rules that promote data back up to a faster tier after previously being migrated down

In general, slower tiers can impact the performance of faster tiers, especially with I/O sensitive workflows:

- Very fast to fast works well:
For example: PowerScale F600 to Isilon H600.
- Streaming or mid-performance to Archive works well:
For example: Isilon H500 to Isilon A200.

- Archive to deep Archive works well:
For example: Isilon A200 to Isilon A2000.

For optimal large cluster performance, Dell EMC recommends observing the following OneFS SmartPools best practices:

- Ensure that cluster capacity utilization (HDD and SSD) remains below 85% on each pool.
- If the cluster consists of more than one node type, direct the default file pool policy to write to the higher performing node pool. Data can then be classified and down-tiered as necessary.
- A file pool policy can have three 'OR' disjunctions and each term joined by an 'OR' can contain at most five 'AND's.
- Define a performance and protection profile for each tier and configure it accordingly.
- File pool policy order precedence matters, as the policies are applied on first match basis (i.e., the first file pool policy to match the expression will be the applied policy).
- By default, the SmartPools job runs only once per day. If you create a file pool policy to be run at a higher frequency, ensure the SmartPools job is configured to run multiple times per day.
- Enable SmartPools Virtual Hot Spares with a minimum of 10% space allocation. This ensures that there's space available for data reconstruction and re-protection in the event of a drive or node failure, and generally helps guard against file system full issues.
- Avoid creating hard links to files which will cause the file to match different file pool policies
- If node pools are combined into tiers, the file pool rules should target the tiers rather than specific node pools within the tiers.
- Avoid creating tiers that combine node pools both with and without SSDs.
- The number of SmartPools tiers should not exceed 5.
- Where possible, ensure that all nodes in a cluster have at least one SSD, including nearline and high-density nodes.
- For performance workloads, SSD metadata read-write acceleration is recommended. The metadata read acceleration helps with getattr, access, and lookup operations while the write acceleration helps reduce latencies on create, delete, setattr, mkdir operations. Ensure that sufficient SSD capacity (6-10%) is available before turning on metadata-write acceleration.
- If SmartPools takes more than a day to run on OneFS 8.2 or later, or the cluster is already running the FSAnalyze job, consider scheduling the FilePolicy (and corresponding IndexUpdate job) to run daily and reducing the frequency of the SmartPools job to monthly. The following table provides a suggested job schedule when deploying FilePolicy:

Job	Schedule	Impact	Priority
FilePolicy	Every day at 22:00	LOW	6
IndexUpdate	Every six hours, every day	LOW	5
SmartPools	Monthly – Sunday at 23:00	LOW	6

- Avoid using the 'isi set' command or the OneFS Filesystem Explorer to change file attributes, such as protection level, for a group of data. Instead use SmartPools file pool policies.

As an extension to tiering, CloudPools enables cluster data to be archived to cloud storage using the same file pool policy engine as SmartPools. Supported cloud providers include Microsoft Azure, Amazon S3, DELL EMC ECS, and native OneFS.

① For large clusters, CloudPools can be used to reduce node pool percentage utilization by transferring cold archive data to the cloud.

📖 More information on OneFS data tiering and file pool policies is available in the [SmartPools white paper](#).

Multi-tenant Recommendations

Within a large cluster, OneFS Access Zones can be configured to provide secure, isolated storage pools. Each division within an organization, for example, can have their own separate zone, while allowing consolidation of storage resources without compromising security.

A cluster includes a built-in access zone, the System zone, where you manage all aspects of a cluster and other access zones. By default, all cluster IP addresses connect to the System zone. Even if you create additional access zones, you configure all access zones in the System zone.

The best practices for Access Zones include:

- The number of access zones should not exceed 50. The number of local users and groups per cluster should not exceed 25,000 for each.
- Isolate corporate tenants with Access Zones, up to a maximum of 50 zones.
- Use the system access zone for cluster management.
- Constrain different protocols (for example, NFS, SMB) to separate access zone
- If you create access zones, ensure that the directory paths for each zone under /ifs do not overlap. Instead, you should designate separate directory trees for each zone.
- OneFS 8.0 and later includes the 'groupnet' networking object as part of the support for multi-tenancy. Groupnets sit above subnets and pools and allow separate Access Zones to contain distinct DNS settings.

① A minimum of two AD, LDAP or NIS servers provides redundancy and helps avoid access control lookups becoming a bottleneck. For larger environments, scaling the number of domain servers may be required.

📖 For more information on identity management, authentication, and access control in combined NFS and SMB environments, please refer to the [OneFS Multiprotocol Security Guide](#).

Job Engine Recommendations

In a OneFS powered cluster, there are jobs that are responsible for taking care of the health and maintenance of the cluster itself. These jobs are all managed by the OneFS job engine. The Job Engine runs, or maps, jobs across the entire cluster and reduces them into smaller work items, which are allocated to multiple worker threads on each node. Jobs are typically executed as background tasks across the cluster, using spare or especially reserved capacity and resources.

With large clusters, there is obviously a high degree of opportunity for job parallelization across a high node count. The flip side to this is that there is also an increased management overhead as a result of coordinating tasks across a large distributed infrastructure. There is also the need to deal in the background with increased levels of hardware transience, failure, and repair. As such, the Job Engine relies upon fine grained impact measurement and management control in order to avoid significantly impacting the core workflows.

The Job Engine jobs themselves can be categorized into three primary classes:

Job Category	Description
File system maintenance	These jobs perform background file system maintenance, and typically require access to all nodes. These jobs are required to run in default configurations, and often in degraded cluster conditions. Examples include file system protection and drive rebuilds.
Feature support jobs	The feature support jobs perform work that facilitates some extended storage management function, and typically only run when the feature has been configured. Examples include deduplication and anti-virus scanning.
User action jobs	These jobs are run directly by the storage administrator to accomplish some data management goal. Examples include parallel tree deletes and permissions maintenance.

Figure 20: OneFS Job Engine Job Classes

The Job Engine allows up to three jobs to be run simultaneously. This concurrent job execution is governed by the following criteria:

- Job Priority
- Exclusion Sets - jobs which cannot run together (i.e., FlexProtect and AutoBalance)
- Cluster health - most jobs cannot run when the cluster is in a degraded state.

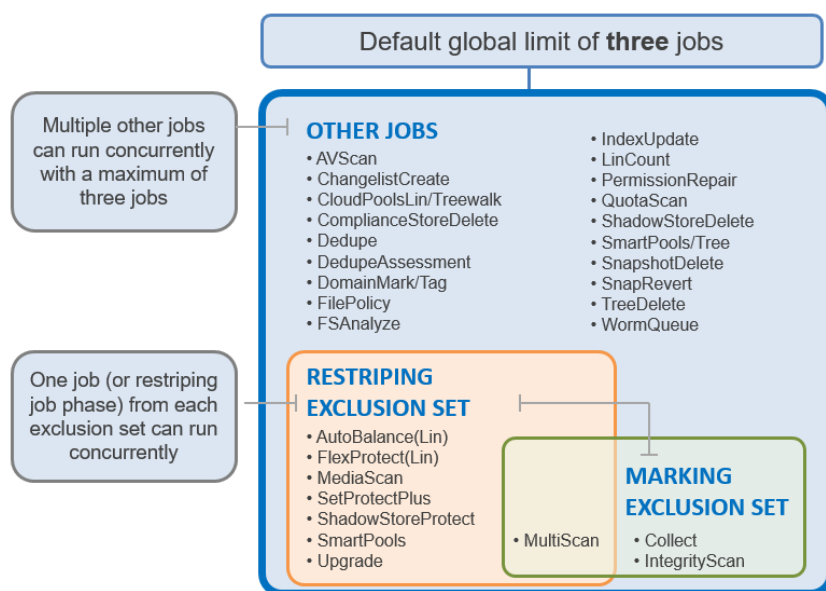



Figure 21: Job Engine Exclusion Sets.

The default, global limit of 3 jobs does not include jobs for striping or marking; one job from each of those categories can also run concurrently.

For optimal cluster performance, Dell EMC recommends observing the following Job Engine best practices:

- Schedule regular maintenance jobs to run during the cluster's low usage hours – overnight, weekends, etc.
- Maintain 15% cluster free space due to the amount of work that SmartPools (if licensed) will do to rebalance data. If SmartPools is unlicensed, SetProtectPlus will be run instead.
- Where possible, use the default priority, impact and scheduling settings for each job.
- When reconfiguring the default priority, schedule and impact profile for a job, consider the following questions:
 - What resources am I impacting?
 - What would I be gaining or losing if I re-prioritized this job?
 - What are my impact options and their respective benefits and drawbacks?
 - How long will the job run and what other jobs will it contend with?
- In a large heterogeneous cluster, tune job priorities and impact policies to the level of the lowest performance tier.
- To complement the four default impact profiles, create additional profiles such as "daytime_medium", "after_hours_medium", "weekend_medium", etc, to fit specific environment needs.
- Ensure the cluster, including any individual node pools, is less than 85% full, so performance is not impacted and that there is always sufficient space to re-protect data in the event of drive failures. Also enable virtual hot spare (VHS) to reserve space in case you need to smartfail devices.
- Configure and pay attention to alerts. Set up event notification rules so that you will be notified when the cluster begins to reach capacity thresholds, etc. Make sure to enter a current email address to ensure you receive the notifications.
- Recommend running MediaScan run to completion before upgrading. However, given the number of drives in a typical large cluster, MediaScan will take a significant duration, so budget time appropriately.
- By default, FlexProtect is the only job allowed to run if a cluster is in degraded mode. Other jobs will automatically be paused and will not resume until FlexProtect has completed and the cluster is healthy again.

- Restriping jobs only block each other when the current phase may perform restriping. This is most evident with MultiScan, whose final phase only sweeps rather than restripes. Similarly, MediaScan, which rarely ever restripes, is usually able to run to completion more without contending with other restriping jobs.
- FlexProtect can take longer than expected on large clusters - typically up to a day or more. SmartFailing and replacing entire nodes can also be a very lengthy process.
- If FlexProtect is running, allow it to finish completely before powering down any node(s), or the entire cluster. While shutting down the cluster during restripe won't hurt anything directly, it does increase the risk of a second device failure before Flexprotect finishes re-protecting data.
- If you need to delete snapshots and there are down or smartfailed devices on the cluster, or the cluster is in an otherwise "degraded protection" state, contact Dell EMC Technical Support for assistance.
- SyncIQ, the OneFS replication product, does not use job engine. However, it has both influenced, and been strongly influenced by, the job engine's design. SyncIQ also terms its operations "jobs", and its processes and terminology bear some similarity to job engine. The job engine impact management framework is aware of the resources consumed by SyncIQ, in addition to client load, and will throttle jobs accordingly.

 Further information is available in the [OneFS Job Engine](#) white paper.

Job Engine Monitoring

Monitoring job performance and resource utilization is critical for the successful administration and troubleshooting of large clusters. A variety of Job Engine specific metrics are available via the OneFS CLI, including per job disk usage, etc. For example, worker statistics and job level resource usage can be viewed with CLI command 'isi job statistics list'. Additionally, the status of the Job Engine workers is available via the OneFS CLI using the 'isi job statistics view' command.

Job events, including pause/resume, waiting, phase completion, job success, failure, etc, are reported, plus a comprehensive job report is also provided for each phase of a job. This report contains detailed information on runtime, CPU, drive and memory utilization, the number of data and metadata objects scanned, and other work details or errors specific to the job type. While a job is running, an Active Job Details report is also available. This provides contextual information, including elapsed time, current job phase, job progress status, etc.

For inode (LIN) based jobs, progress as an estimated percentage completion is also displayed, based on processed LIN counts.

Detailed, granular job performance information and statistics are available in a job's report. These include per job phase CPU and memory utilization (min, max and average), and total read and write IOPS and throughput.

In addition to detailed NFS, SMB and S3 protocol and workflow breakdowns, OneFS also includes a real-time job performance resource monitoring framework, which provides statistics for the resources used by jobs - both cluster-wide and per-node. This information is provided via the isi statistics workload CLI command. Available in a 'top' format, this command displays the top jobs and processes, and periodically updates the information.

For example, the following syntax shows, and indefinitely refreshes, the top five processes on a cluster:

```
# isi statistics workload --limit 5 --format=top
```

```
last update: 2018-06-11T06:45:25 (s)ort: default
```

CPU	Reads	Writes	L2	L3	Node	SystemName	JobType
1.4s	9.1k	0.0		3.5k	497.0	2 Job: 237	IntegrityScan[0]
1.2s	85.7	714.7		4.9k	0.0	1 Job: 238	Dedupe[0]
1.2s	9.5k	0.0		3.5k	48.5	1 Job: 237	IntegrityScan[0]
1.2s	7.4k	541.3		4.9k	0.0	3 Job: 238	Dedupe[0]
1.1s	7.9k	0.0		3.5k	41.6	2 Job: 237	IntegrityScan[0]

The resource statistics tracked per job, per job phase, and per node include CPU, reads, writes, and L2 & L3 cache hits. Unlike the output from the 'top' command, this makes it easier to diagnose individual job resource issues, etc.


Data Availability, Protection & Disaster Recovery Considerations

At the core of every effective large cluster data protection strategy lies a solid business continuance plan. An explicitly defined and routinely tested procedure is key to minimizing the potential impact to the workflow when a failure occurs or in the event of a natural

disaster. There are a number of ways to address data protection and most enterprises adopt a combination of these methods, to varying degrees. Among the primary approaches to data protection at scale are fault tolerance, redundancy, snapshots, and replication. Some of these methods are biased towards cost efficiency but have a higher risk associated with them, and others represent a higher cost but also offer an increased level of protection.


 More information is available in the [Data Availability and Protection](#) paper

① Despite support for parallel NDMP and native 2-way NDMP over fibre channel, traditional backup to VTL or tape is often not a feasible DR strategy at the large or extra-large cluster scale. Instead, replication is usually the preferred tool of choice.

 Further information is provided in the [Backup and Recovery](#) guide.

For large clusters, snapshots typically provide the first line of defense in a data protection strategy with low recovery objectives. OneFS SnapshotIQ affords the following benefits:

- Snapshots are created at the directory-level instead of the volume-level, thereby providing improved granularity.
- There is no requirement for reserved space for snapshots in OneFS. Snapshots can use as much or little of the available file system space as desirable.
- Integration with Windows Volume Snapshot Manager
- Snapshot creation, restoration and deletion is easily managed using flexible policies and schedules.
- Using SmartPools, snapshots can physically reside on a different disk tier than the original data.

 Further information is available in the [OneFS SnapshotIQ](#) white paper.

Replication Recommendations


Data protection and disaster recovery at the large cluster level is another area that relies upon a well-designed architecture. For example, many is the cluster that has started out with NDMP backup as the DR strategy and grown organically into the multi-petabyte size. At this point, realizing that traditional backup is not scaling, replication has had to be hurriedly introduced – often involving a lengthy initial replication job for a huge dataset over a WAN link.

OneFS SyncIQ provides high-performance, asynchronous replication of unstructured data to address a broad range of recovery objectives which scale linearly up into the large and extra-large cluster ranges. Utilizing a highly-parallel, policy-based replication architecture, SyncIQ works across either LAN or WAN connectivity, providing protection from both site-specific and regional disasters.

Synchronization policies may be configured at the file, directory, or entire file system-level and have a priority setting to allow favored policies to preempt others. In addition to chronological scheduling, replication policies can also be configured to start whenever the source is modified (change based replication).

Leveraging SnapshotIQ, the Linear Restore functionality of SyncIQ is able to detect and restore (commit) consistent, point in time, block-level changes between cluster replication sets, with a minimal impact on operations and a granular RPO. In the event that a primary cluster becomes unavailable, SyncIQ provides the ability to failover to a mirrored, DR cluster. During such a scenario, the administrator makes the decision to redirect client I/O to the mirror and initiates SyncIQ failover on the DR cluster. Users will continue to read and write to the DR cluster while the primary cluster is repaired. As such, disaster recovery sites for large clusters typically contain SyncIQ target cluster(s) which store replicated data from the primary site(s).

- The recommended limit of running SyncIQ policies is 1000 policies and 50 concurrent jobs per cluster.
- While the maximum number of workers per node per policy is eight, the default and recommended number of workers per node is three. The recommended limit of workers per replication policy is 40.
- Consider using SmartConnect pools to constrain replication to a dedicated set of cluster network interfaces, and to avoid contention with other workflows accessing the cluster through these nodes.
- Use SyncIQ network throttling to control how much network bandwidth SyncIQ can consume.
- Periodic testing of SyncIQ failover is recommended to ensure that the data on DR site is available as desired.
- If a cluster is running OneFS 8.2 or later, use SyncIQ encryption to protect any replication sessions that traverse WAN or other insecure or untrusted network segments.

 Further information is available in the [OneFS SyncIQ](#) white paper.

Best Practices Checklist

For optimal large cluster operation and performance, Dell EMC recommends observing the following best practices. Please note that this information will likely be covered elsewhere in this paper.

- ✓ When it comes to architecting and scaling large OneFS powered clusters, plan ahead and strive for simplicity.
- ✓ Just because you can build a large cluster doesn't necessarily mean you should.
- ✓ For high performance workloads, consider a pod architecture.
- ✓ Undertake a thorough analysis of any the workloads and applications that any large cluster will be supporting.
- ✓ Define, implement and regularly test a data protection and recover strategy and business continuance plan.
- ✓ Maintain sufficient free space and pay attention to data ingest rate. Keep cluster capacity utilization (HDD and SSD) below 85%.
- ✓ Ensure Virtual Hot Spare and SmartPools spillover both remain enabled (the default).
- ✓ For large PowerScale and Isilon Gen6 clusters, use an Ethernet back end network whenever possible.
- ✓ Connect all nodes to a front-end Ethernet network (avoid NANON).
- ✓ If SmartPools is licensed, ensure that spillover is enabled (default setting).
- ✓ Implement and maintain a thorough cable coloring, naming and management convention.
- ✓ Keep a cluster maintenance and change log.
- ✓ The key for file and directory layout always revolves around balance. Keep the directory tree structure and its file contents as uniform as possible.
- ✓ Manage your data: Archive infrequently accessed data and delete unused data.
- ✓ Maintain appropriate data protection levels as the cluster grows
- ✓ Record your original settings before making any configuration changes to OneFS or its data services.
- ✓ Monitor cluster capacity and data ingest rate.
- ✓ Ensure that all desired data services are licensed and configured.
- ✓ Observe NFS and SMB connection limits.
- ✓ Many cluster configuration settings are global and have cluster-wide effects. If you consider changing cluster-wide configuration settings, be sure that you fully understand the global settings and their implications
- ✓ Manage snapshot creation and deletion schedules.
- ✓ Setup SmartConnect for load balancing and use Round Robin as the balancing policy.
- ✓ Use SmartQuotas to understand, predict, control and limit storage usage.
- ✓ Avoid running Job Engine jobs at 'HIGH' impact on large clusters.
- ✓ If using SmartPools tiering, reconfigure the Storage Target field from "anywhere" to a specific tier or node pool to direct ingest to a performance node pool or tier.
- ✓ Ensure the SmartPools job only runs during off-hours.
- ✓ Add cluster to an InsightIQ monitoring instance, assuming the cluster is no more than 80 nodes in size.
- ✓ Deploy a lab cluster or [OneFS Simulator](#) environment to test and validate any new cluster configurations before making changes that affect the production environment.
- ✓ Confirm that remote support functions work correctly through EMC Secure Remote Support (ESRS) and internal email/SNMP notifications.
- ✓ Upgrade OneFS to a newer release at least once a year.
- ✓ Configure and pay attention to cluster events and alerts and/or monitor log files and SNMP MIBs and traps.
- ✓ Regularly run and review cluster health check reports.

- ✓ Keep node and drive firmware as up to date as possible. This is especially important with Isilon Gen6 hardware.
- ✓ Sign up for product updates on the [Dell EMC support site](#) for notification on ETAs, KBs, new releases, breaking issues, etc.
- ✓ While the best practice is to keep clusters up-to-date on OneFS releases, when that's not possible at least look at new the versions' release notes to determine if they contain any bug fixes pertinent to your workflow
- ✓ Capacity planning: Use FSA and IIQ tools to plan ahead and never let your cluster get too full given the performance impacts (and recovery efforts required) you could run into.
- ✓ Ensure that alerting is properly configured and that not only you (and your team) are receiving connect homes, but that Dell EMC is as well. When support cases are opened, we take quick action on them even when you're busy.
- ✓ Utilize ESRS for log uploads, remote access, etc.
- ✓ If you've got multiple node types/generations, choose wisely on where to tier data to avoid SmartPools job impacts, or impacts from tier capacity imbalance.
- ✓ Your cluster can be impacted by backend network issues: Keep up to date on Infiniband issues and firmware and leverage the 'ibstat' CLI command, as it can be useful to find less than performant cabling in your environment.
- ✓ Ensure your nodes are running well within temperature and humidity tolerances, not on the edges.

Summary

Dell EMC PowerScale overcomes the problems that undermine traditional NAS systems by combining the three traditional layers of storage architecture—file system, volume manager, and data protection—into a scale-out NAS cluster with a distributed file system. The Dell EMC PowerScale scale-out architecture eliminates controller bottlenecks to reduce wall-clock runtimes for concurrent jobs, accelerates metadata operations, improves storage efficiency to lower capital expenditures, centralizes management to reduce operating expenses, and delivers strategic advantages to increase competitiveness in a global market

TAKE THE NEXT STEP

Contact your Dell EMC sales representative or authorized reseller to learn more about how Dell EMC PowerScale scale NAS storage solutions can benefit your organization.

[Visit Dell EMC PowerScale](#) to compare features and get more information.



Learn more about Dell EMC PowerScale solutions



Contact a Dell EMC Expert



View more resources



Join the conversation with #DellEMCStorage