Technical White Paper

Dell EMC PowerScale: Autodesk Flame 2022 Best Practices and Configuration Guide

Abstract

This paper describes best practices for using Dell EMC PowerScale™ OneFS™ 9.x storage with Autodesk™ Flame 2022™.

September 2021

Revisions

Date	Description
March 2019	Initial release
September 2019	Update for Flame 2020
September 2021	Update for Flame 2022

Acknowledgments

Author: Gregory Shiff, Principal Solutions Architect, Media and Entertainment

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.

Table of contents

Executive summary

This document outlines the interaction between PowerScale OneFS storage and Autodesk Flame running on CentOS. The paper provides guidance on achieving the best performance with Flame.

The storage qualification testing procedure was undertaken with PowerScale F800 All Flash, OneFS 9.2, and Autodesk Flame 2022 in the Dell PowerScale M&E Lab, August 2021.

This document does not make specific recommendations regarding component hardware, nor does it make any claims regarding specific performance.

This document consists of the following sections:

- Application demands on shared storage -- Some general information about the unique demands video applications and Flame put on storage
- PowerScale optimization guidelines -- Specific recommendations for configuring an PowerScale cluster to perform optimally with Flame
- Flame 2022 guidelines -- Environment variables to tune Flame performance with PowerScale
- Reference architecture -- Equipment used for certification of PowerScale F800 and Flame 2022

1 Application demands and general guidelines

The demands of editing, compositing, 3D effects, and finishing are challenging - particularly at high frame rates and resolutions. Autodesk Flame relies on a performant workstation, networking, and storage infrastructure. When designing a system, architects must consider both quantitative and subjective requirements, such as:

- Throughput
- Latency
- Demands of different codecs
- Scrubbing
- Multitrack
- Subjective “feel”

As part of the Autodesk qualification program, the Dell 7920 Precision Workstation and PowerScale F800 storage are qualified to support Autodesk Flame. The PowerScale F600 and F900 all-flash platforms also provide excellent performance for Flame workflows.

You can find additional information about the system requirements of Autodesk Flame at:

https://knowledge.autodesk.com/support/flame-products/learn-explore/caas/sfdcarticles/sfdcarticles/flame-sysreqs.html

1.1 Overview of the benefits of using Dell EMC PowerScale

Using Dell EMC PowerScale brings significant benefits to Autodesk Flame workflows and environments:

- Dell EMC PowerScale is easily scalable - and can grow with your business.
- You can eliminate silos of task-specific storage. You can work directly from the centralized PowerScale storage repository without needing to move media around in a “push-pull” workflow.
- You can maintain a single namespace across tiers of storage - including performant Dell EMC PowerScale H500 and F800, and (for example) file archive Dell EMC Isilon A2000.

1.2 Overview of file types

You can use Autodesk Flame with various file types:

- Metadata
  - Action Only Formats
  - Sequence Exchange Formats
- Media
  - Image Sequences
  - Clip-Based Media

For more information about file types and formats supported by Autodesk Flame, see the AutoDesk Flame 2022 Documentation.

Metadata files

Metadata and sequence information files are typically relatively small, but frequently updated. These files contain information about the sequences being created or manipulated. For example, such a file could contain information about which clips are used in a sequence, or about how the colors should be corrected.

Although metadata and sequence information files are frequently accessed and updated, they should not contribute significantly to system throughput because they are small.

Media files

Because of their large size, media files of encoded video can represent the most important challenge to an architect when designing systems to support creative applications. Video can be encoded for digital storage either as an image sequence, or as a self-contained clip.

Image sequences

An image sequence represents each frame of video with a separate file. The files for a sequence are stored in the same directory, and have a fixed base name with a decimal or hexadecimal incrementing suffix. Autodesk Flame works with several different image sequence formats, including:

- DPX
- ARRI
- Open EXR

Different applications use different naming structures and standards within this form, but some illustrative examples for DPX sequences would be:

```
filebasename_scene-01_00056.dpx
filebasename_scene-01_00057.dpx
filebasename_scene-01_00058.dpx

filebasename_scene-05_0FC09.dpx
filebasename_scene-05_0FC0A.dpx
filebasename_scene-05_0FC0B.dpx
```

For more information about these formats, including their respective strengths and use-cases, see:

https://en.wikipedia.org/wiki/Digital_Picture_Exchange
http://www.arri.com/camera/alexa/workflow/working_with_arriraw/arriraw/format/
http://www.openexr.com
https://en.wikipedia.org/wiki/OpenEXR

Clip-based media

A clip-based media file represents an entire sequence of video frames (and often interleaved audio, time code, and metadata) in a single file on the storage. Autodesk Flame works with several different clip-based codecs, including:

- RED -- https://www.red.com/red-101
- XAVC -- https://en.wikipedia.org/wiki/XAVC
- ProRes -- https://en.wikipedia.org/wiki/Apple_ProRes

- DNX -- https://en.wikipedia.org/wiki/Avid_DNxHD and http://www.avid.com/products/avid-dnxhr-and-dnxhd

Compression

You can compress both image sequences and clip-based media. Compression can yield benefits in required storage capacity and throughput but it can also generate computational overhead and reduce the perceived visual and audio quality of the media.

Compression can be lossless (where the exact original image data can be recovered) or lossy (where some image data is discarded in the compression process). Some lossy compression is “visually lossless” (where some image data is discarded, but in a way that is not detectable by the human eye).

1.3 Performance metrics

Users of Autodesk Flame often work with uncompressed image sequences. It is important to design a full system that can support the high-performance demands of these workflows.

Because Autodesk Flame media makes the highest demands of workstations, networking, and storage in any implementation, systems must be architected to accommodate those performance requirements.

The two key system performance metrics that architects must consider when specifying storage solutions for Autodesk Flame are *throughput* and *latency*.

Throughput

Throughput performance is a measure of the required *sustained* data transfer rate between the storage and the application. To sustain playback at the application layer, the system must be able to support at least the required throughput over an indeterminate duration.

The following table summarizes the five important factors that dictate the size (and resulting throughput requirements) of media files, and their impact on the demands of supporting systems.

Table 1 Throughput factors

Characteristic	Possible Value
Frame Size	SD, HD, 2K (DCI) UHD, 4K (DCI), 8K (DCI)
Frame Rate	24, 25, 30, 50, 60 Frames per Second
Color Depth	10-bit (Standard Dynamic Range), 12-bit (High Dynamic Range), 16-bit (High Dynamic Range)
Encoding Method	Image Sequence or Clip-Based
Compression Method	Uncompressed, Lossless, Visually Lossless, Lossy

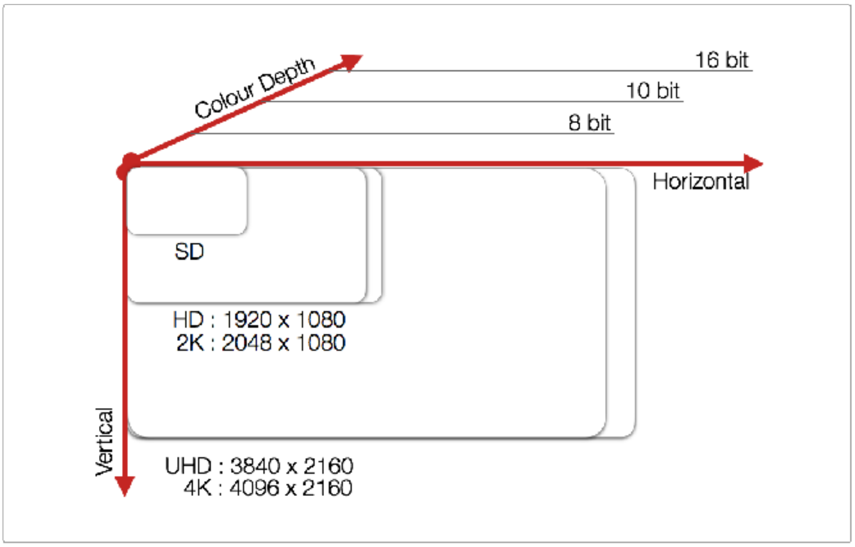


Figure 1 Relative frame size and color depth

Latency

Latency performance is a measure of the required *instantaneous* data transfer rate. To offer a responsive experience at the application layer, the system must be able to deliver frames in a timely manner after they have been requested. One measure of latency is the *time to first byte* - how long it takes data to start flowing once the application requests it.

Flame will request each frame sequentially while the preceding frame is being displayed in image sequence workflows (where each frame of video is stored as a separate file). It follows that for steady and sustained playback, each file must be requested and retrieved in less than the time for which each frame is displayed.

Depending on the frames per second, the rate of each frame can range from 42 milliseconds for 24 fps down to 17 milliseconds for 60 fps material.

1.4 Basic infrastructure guidelines

Workstation specification

Dell Technologies recommends a suitably specified Precision Workstation 7920 or later for use with Autodesk Flame. Specification and configuration of the workstation is outside of scope of this document. For more information, see Flame Family 2022 release system requirements.

PowerScale and OneFS version

PowerScale F800 All Flash is qualified to support Autodesk Flame 2022 environments. OneFS version 9.2 has been tested and qualified. Later versions of OneFS may deliver equivalent or improved performance.

Network connectivity

To achieve the throughput necessary for typical Autodesk Flame workflows, a full 40 GbE or higher network is required, including PowerScale storage, network infrastructure, and client interface.

The system used while preparing this document was configured to use Jumbo Frames (MTU 9000).

File sharing configuration

In a collaborative environment, multiple users, using multiple workstations are required to access and share files. Standard guidance when using any distribution of Linux is to use the NFS file sharing protocol. We assume that the most appropriate variant of the protocol for most environments is NFSv3.

You can use the administrative web interface of OneFS to create an NFS Export on the PowerScale storage cluster. For details, see Dell EMC PowerScale: OneFS Design Considerations and Best Practices.

Care should be taken to architect a OneFS NFS export and permissions strategy that accommodates the required workflow. Configuring OneFS NFS exports and designing a permissions strategy is outside of the scope of this document. For detailed information, see NFS Design Considerations and Best Practices Whitepaper.

2 PowerScale OneFS optimization guidelines

Architects should consider four key optimizations for OneFS when deploying in an Autodesk Flame environment.

- Metadata Read/Write Acceleration
- Streaming and Concurrency Modes
- Filename Prefetch
- NFS over RDMA

2.1 Metadata read/write acceleration

File access performance can be optimized by enabling Metadata read/write acceleration. When enabled, this optimization uses SSDs for reading and writing file system metadata, which improves access time and reduces latency. Where all underlying storage units are SSD, such as the PowerScale F800 All Flash, then this setting is not a meaningful or required optimization.

We performed the qualification for Flame 2022 using PowerScale F800 All Flash. Previous versions of Flame were qualified against an PowerScale H600 node with 10K SAS drives and SSDs. When using the H600 node with Flame, this optimization is relevant.

Enabling metadata read/write acceleration

You can use the OneFS Web User Interface to enable metadata read/write optimization as a File Pool Policy. For more information and guidance, see the published documentation: OneFS 9.2 SSD Strategies.

2.2 Streaming and concurrency modes

OneFS gives control over how data is laid out and accessed from disk. These modes can be set at the directory level, allowing administrators to select the appropriate layout and access strategies for the files within a particular path. The default mode for file layout and access is Concurrency, which is appropriate for most workloads (even media workloads). However, when serving up high throughput media files, Streaming Mode is likely the most appropriate.

Streaming mode

You can optimize playback of high throughput media from OneFS by enabling Streaming Mode at the file pool, or at the directory level. Streaming performance is important when playing media linearly, such as when playing a video sequence in a time line.

Before enabling Streaming Mode, consider the other workflows hosted in the same pool or directory. You may be able to achieve the best performance for all workflows on a single cluster by selectively enabling Streaming Mode on a limited number of directories.

Streaming Mode optimizes two behaviors of OneFS to deliver increased streaming performance:

- Data is striped across more underlying storage units (disk drives or SSDs).
- Data is prefetched aggressively.

During the Flame 2022 Storage Qualification tests, Streaming Mode improved playback performance of high data-rate codecs.

Concurrency mode

Due to Streaming Mode's aggressive prefetching strategy, there are circumstances when Flame may start to drop frames due to excessive prefetching from multiple simultaneous clients. In those circumstances, it may be preferable to enable Concurrency Mode on the directory that contains the media.

In general, administrators should set media directories to Streaming Mode. If Flame drops frames when the storage should otherwise support playback, switching the media directories to Concurrency Mode is a relatively quick change to determine whether there is a positive (or negative) impact. In these circumstances, monitor the workstation performance closely, because it is more likely the cause of poor playback performance.

Choosing streaming mode vs concurrency mode

Streaming Mode is optimized for a few clients playing back high throughput material such as ProRes 4444 4K @ 60 fps. Concurrency mode uses an adaptive prefetch algorithm that is more measured in its approach. For clusters with large numbers of users working with relatively low-bandwidth material (such as ProRes 422 1080p @ 30 fps), sticking with Concurrency Mode is the most appropriate choice.

A strength of OneFS is that both streaming and concurrency mode can co-exist on the same cluster on a directory by directory basis. A common approach is to use Concurrency Mode generally while selectively setting media directories with high throughput material to Streaming Mode.

Enabling streaming mode or concurrency mode

You can use the OneFS Web User Interface to enable Streaming or Concurrency mode as a File Pool Policy. You can also use the `isi set` CLI command to configure these modes. More information and guidance are available in the following published documentation:

OneFS web administration: Configure default I/O optimization settings

OneFS web administration: Modify file and directory properties

OneFS command line administration

Note: When filename prefetch (outlined below) is also required and enabled, there is an interaction between the two optimizations.

2.3 Filename prefetch

You can enable filename prefetch to optimize the streaming performance of OneFS when working with image sequences. It is critical that this optimization only be applied to directories that contain image sequences of successively named files. Applying it to directories that do not contain image sequences can result in *false fetches* (where resources are used to prefetch unneeded files).

When Flame is used with image sequences, you may achieve improved performance by enabling filename-based prefetch. If image sequences are not used in the environment, there is no need for filename-based prefetch.

While running the Flame 2022 storage qualification tests involving image sequences, we achieved significant playback performance improvement with filename prefetch enabled.

Operation

Filename prefetch enables performance optimization by detecting when a client requests image sequences. When a sequence is detected, OneFS prefetches files from the underlying storage units before they are requested.

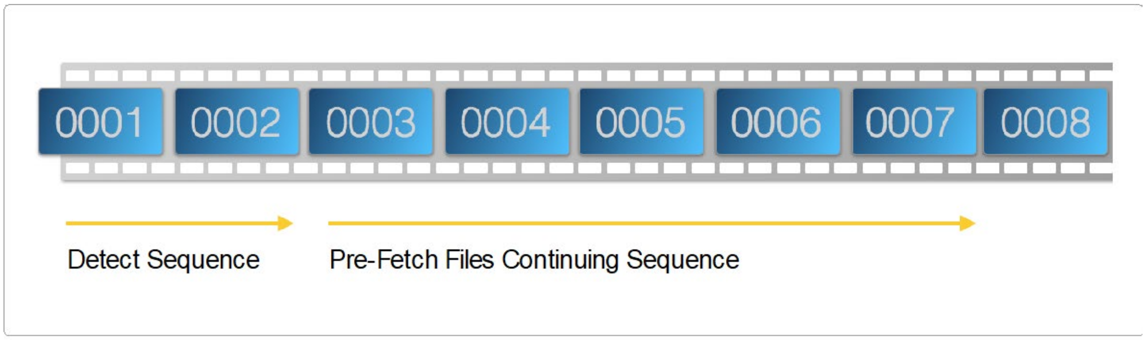


Figure 2 FileName prefetch example

The filename prefetch algorithm is optimized to detect image sequences with either decimal or hexadecimal filename numerical increments.

Enabling

You can use the OneFS command line to enable Filename Prefetch on individual directories. For configuration details, see the Filename Based Pre-Fetch Guide. This also describes how to set and examine OneFS file layout and access strategies using the command-line interface. This paper is recommended reading even if filename prefetch is not to being used. The paper references OneFS 8 but is still relevant to OneFS 9.x.

Additional information

For detailed guidance on file pool policies, data layout, and access strategies, see:

PowerScale OneFS Technical Overview

For guidance for using the command line to make administrative changes, see:

PowerScale OneFS CLI Administration Guide

For guidance for using the web interface to make administrative changes, see:

OneFS Web Administration Guide

2.4 NFS over RDMA

OneFS 9.2 introduces support for NFS over RDMA. This new feature provides better storage performance, reduced latency, and reduced CPU load. These benefits are intimately related when it comes to Flame.

When playing back high throughout media, such as uncompressed 8K DPX or EXR, the TCP protocol itself becomes a bottleneck. At a high level, RDMA eliminates this bottleneck by allowing data to be transferred directly to the memory of the workstation without involving the CPU directly, offloading these cycles to the network interface card. The degree of improvement depends on the data being read, workstation performance, and the load on the PowerScale storage.

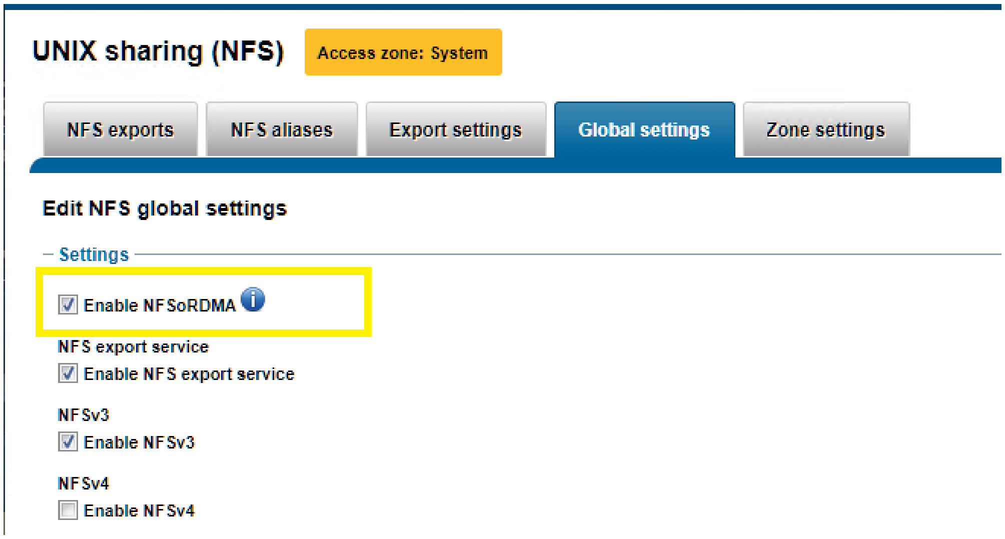
In media playback testing with RDMA, mounting the PowerScale storage using RDMA (as opposed to TCP) yielded as much as 50-100% better read throughput. But there is a caveat. Most of the time, workstation performance is the limiting factor before storage performance. This caveat was evidenced by testing performed for this white paper. When the workstation was working hard to decode heavily compressed, high-resolution media, the throughput gains were more modest. The good news is that RDMA also improves workstation performance.

When accessing data from network storage (such as PowerScale OneFS), there is workstation CPU overhead to assemble the incoming packets and make sense of that data. As mentioned above, RDMA reduces this CPU load (RDMA also reduces the CPU load on the PowerScale cluster), by offloading cycles to the network card. These extra CPU cycles free up the workstation to focus on media operations.

The benefits of RDMA were evident when playing back a 4K DPX sequence at 60 frames-per-second. Enabling debug mode in Flame (shift+D+insert) shows a display of GPU dropped frames, broadcast output dropped frames, and video disk dropped frames. With both TCP and RDMA mounts, the “video disk” never dropped a frame, because PowerScale performance is fast enough to play back this material. However, with the storage mounted using TCP, both the GPU and broadcast output showed a significant number of dropped frames because the workstation was struggling to keep up. With the volume mounted using RDMA, all three indicators showed no dropped frames. The RDMA-mounted PowerScale had reduced the workstation load enough that it was now able to properly output the video.

Enabling NFS over RDMA

Enabling RDMA in OneFS is as simple as checking the “Enable NFSoRDMA” box in **Protocols → UNIX sharing (NFS) → Global settings**. This setting allows client systems to mount the NFS shares using RDMA.


UNIX sharing (NFS)

Edit NFS global settings

Settings

☑ Enable NFSoRDMA

NFS export service

☑ Enable NFS export service

NFSv3

☑ Enable NFSv3

NFSv4

□ Enable NFSv4

Figure 3 Enabling NFS over RDMA

There are additional considerations for using RDMA with OneFS on the client and network side. OneFS uses RoCEv2 (RDMA over converged Ethernet) for RDMA. This protocol has its own set of requirements. As of the time of this writing, the only workstation network adapters that OneFS supports for RDMA traffic are Mellanox™ ConnectX-3 Pro™, ConnectX-4™, ConnectX-5™, and ATTO™ FastFrame™ NICs.

Section 6 in the NFS Design Considerations and Best Practices Whitepaper provides details about configuring RDMA in OneFS. A blog post by Dell Technical Marketing also has excellent step-by-step instructions for mounting OneFS using RDMA: Unstructured Data Tips.

3 Autodesk Flame 2022 environment variable optimization

The environment variables noted below change the way that Flame interacts with its host workstation. It may be tempting to increase certain settings to a high level to try to achieve a maximum improvement. However, artificially large settings can have a detrimental effect, such as causing application instability and crashes.

Any changes that are made should be tested thoroughly before deploying in a live environment. We recommend configuring only the minimum deviation required from default values to achieve the required performance increase. Dell does not provide any warranty for setting or changing environment variables.

3.1 Threads for managed and unmanaged partitions

Autodesk Flame by default launches three threads to read files and frames from storage. By increasing the number of threads, you can increase concurrency and therefore improve throughput performance.

Threads for Managed Partition

Use the following environment variable to set the number of threads maintained by the application when reading from a *Managed Partition*:

```
setenv SW_NUM_DISK_IO_PROCS 6
```

Threads for Unmanaged Partition

This change is noted here for completeness, but in tests the changes did not yield a significant performance improvement.

You can use the following environment variable to set the number of threads maintained by the application when reading from an *Unmanaged Partition*:

```
setenv SW_NUM_VIDEO_THREADS 6
```

3.2 Prefetch buffer

In testing, this change yielded some significant performance improvements for general throughput.

Autodesk Flame has an application prefetch buffer. In other words, the application may be configured to request files and frames from storage before they are required - and stage them in workstation RAM.

If the application prefetches more frames to RAM, the environment will better tolerate any transient latency issues.

You can increase the number of frames that the application attempts to keep in its buffer. You can also increase the amount of RAM allocated to the process. In testing, increasing the number of frames had a positive effect on the latency and stability of application playback, resulting in fewer dropped frames.

The default value is 40.

The value is the *total* number of frames across all video tracks being played at that moment.

To increase the number of frames, set the following environment variables:

```
setenv DL_PIPELINE_PREFETCH_POOL_MB 2048
setenv DL_PIPELINE_PREFETCH_POOL_MAX_NB_BUFFERS 80
```

Consideration should be given to the amount of system resources (at the application, client workstation, and storage) consumed by overly aggressive prefetching and the amount of workstation RAM available.

3.3 Editing .cshrc

By default, Flame uses the C-Shell for processing commands. For the above environment variables to persist across reboots (or even new terminal instances), you must add those commands to the C-Shell user profile. The user’s C-Shell profile is stored in the .cshrc file at root of the Flame user’s home directory.

Using a text editor, add the following lines to the bottom of the .cshrc file:

```
setenv SW_NUM_DISK_IO_PROCS 6
setenv SW_NUM_VIDEO_THREADS 6
setenv DL_PIPELINE_PREFETCH_POOL_MB 2048
setenv DL_PIPELINE_PREFETCH_POOLMAX_NB_BUFFERS 80
```

Next, use the following command to source the profile:

```
source ~/.cshrc
```

To verify that the environment variables have been set, run the following command from a new terminal instance and grep for the above environment variables:

```
printenv
```

For example:

```
vxfhost: /opt/Autodesk/flame_2022.0.1/ % source ~/.cshrc
vxfhost: /opt/Autodesk/flame_2022.0.1/ % printenv | grep NUM
SW_NUM_DISK_IO_PROCS=6
SW_NUM_VIDEO_THREADS=6
vxfhost: /opt/Autodesk/flame_2022.0.1/ % printenv | grep PIPELINE
DL_PIPELINE_PREFETCH_POOL_MB=2048
DL_PIPELINE_PREFETCH_POOLMAX_NB_BUFFERS=80
```

4 Architecture used for Flame 2022 storage certification

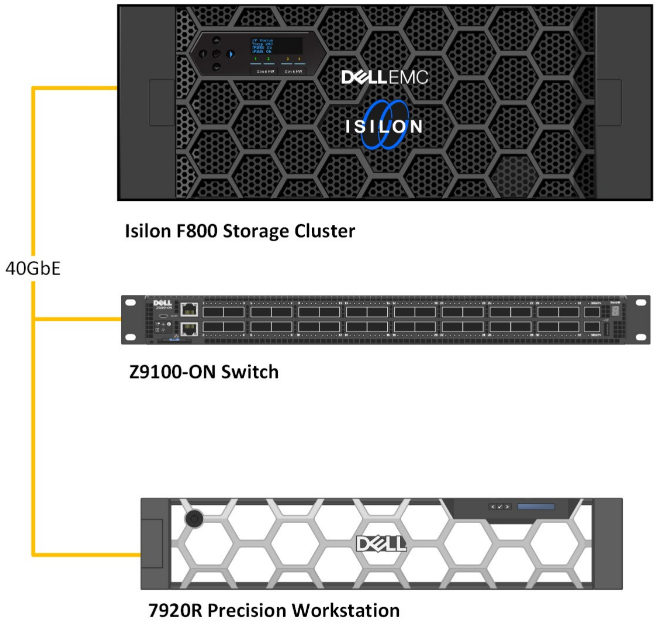


Figure 3 Flame 2022 storage certification architecture

PowerScale storage passed the Autodesk Flame 2022 Storage certification with a PowerScale F800 storage array, Dell Z9100-ON switch, and Dell Precision 7920 rack-mounted workstation. Detailed system requirements for Flame 2022 can be found here:

> https://knowledge.autodesk.com/support/flame-products/troubleshooting/caas/sfdcarticles/sfdcarticles/flame-sysreqs.html

PowerScale F800 storage cluster

- 4-Node PowerScale F800 All Flash Storage Cluster
- 40 GbE Front and back-end networking

Dell Precision 7920R rack mount workstation:

- Dual Xeon Gold 6146 @ 3.2Ghz
- RAM: 768 GB
- GPU: Nvidia GV100
- ATTO 40 GbE NIC
- AJA Kona4 broadcast output card

Dell Z9100-ON network switch

- 40 GbE Networking from PowerScale cluster to workstation

A Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

Storage technical documents and videos provide expertise that helps to ensure customer success on Dell EMC storage platforms.