

Dell EMC Isilon: Autodesk Flame 2020 Best Practices

Abstract

This paper describes best practices for using Isilon storage with Dell EMC Isilon™ OneFS™ 8.x with Autodesk™ Flame 2020.

September 2019

Revisions

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

Acknowledgements

This paper was produced by the following:

Author: Gregory Shiff

Support: Simon Haywood

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 © 2019 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. [8/21/2019] [Technical White Paper] [H16952]

Table of contents

Revisions.....	2
Acknowledgements.....	2
Table of contents	3
Executive summary.....	4
1 Application demands and general guidelines.....	5
1.1 Overview of the benefits of using Dell EMC Isilon.....	5
1.2 Overview of file types	5
1.3 Performance metrics	7
1.4 Basic infrastructure guidelines.....	9
2 Isilon OneFS optimization guidelines	11
3 Autodesk Flame 2020 environment variable optimization	14
3.1 Threads for managed and unmanaged partitions	14
3.2 Prefetch buffer	14
4 Architecture used for Flame 2020 storage certification.....	16
A Technical support and resources	17

Executive summary

This document outlines the interaction between Isilon storage and Autodesk Flame and provides guidance on achieving the best performance.

This document is based on work undertaken with Isilon F800 All Flash, OneFS 8.2, and Autodesk Flame 2020.

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

This document assumes knowledge of the portfolio of Dell Isilon - including H600 and F800. More information is available at:

<https://www.emc.com/collateral/software/data-sheet/h10541-ds-isilon-platform.pdf>

For more background information on working with uncompressed media and image sequences, this document should be read in conjunction with “Uncompressed High-Resolution Workflows with Dell EMC Isilon” H16818.

<https://www.emc.com/collateral/whitepaper/h16818-wp-uncompressed-high-resolution-workflows-isilon.pdf>

This document is broadly broken down into four sections

- Application demands on shared storage
 - Some general information on the unique demands video applications and Flame put on storage
- Isilon optimization guidelines
 - Specific recommendations for configuring an Isilon cluster to perform optimally with Flame
- Flame 2020 guidelines
 - Some environment variable settings that can impact performance of Flame when used with Isilon storage
- Reference architecture
 - Equipment used for certification of Isilon F800 & Flame 2020

1 Application demands and general guidelines

The demands of editing, compositing, 3D effects and finishing are high - particularly at high frame rates and resolutions - and consequently Autodesk Flame relies on a highly performant workstation, networking and storage infrastructure. When designing a system, architects will need to consider both quantitative and subjective requirements.

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

As part of the Autodesk qualification program, the Dell 7920 Precision Workstation and Dell Isilon F800 storage are qualified to support Autodesk Flame.

More information about the system requirements of Autodesk Flame can be found 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 Isilon

The use of Dell EMC Isilon brings significant benefits to Autodesk Flame workflows and environments:

- Dell EMC Isilon is easily scalable - and can grow with your business.
- Silos of task-specific storage may be eliminated. It is possible to work directly from the centralised Isilon storage repository - with no requirement to move media around in a “push-pull” workflow.
- A single namespace may be maintained across tiers of storage - including performant Dell EMC Isilon H600 and F800, and (for example) file archive A2000.

1.2 Overview of file types

Autodesk Flame may be used with a number of different file types:

- Metadata:
 - Action Only Formats
 - Sequence Exchange Formats
- Media:
 - Image Sequences
 - Clip Based Media
 - QuickTime clips
 - Media Exchange Format (MXF) clips
 - MP4 clips
 - Other Streaming Codecs

More information about file types and formats supported by Autodesk Flame can be found at:

<https://help.autodesk.com/view/FLAME/2020/ENU/?guid=GUID-982B4FA6-2DB9-4E0D-ABAA-6AC9FD64568C>

Each file type presents a different challenge to each part of the system, and the file types being used by the required workflow must be considered.

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 information about how the colours should be corrected.

Because metadata and sequence information files are small, they should not present any meaningful contribution to throughput requirements. However, it is important to remember that the files will be frequently accessed and updated.

Media files

Because of their large size, media files of encoded video are normally considered to 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 sequence

An image sequence represents each frame of video with a separate file. The files for a sequence are stored in the same directory, 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
```

More information about these formats - including their respective strengths and use-cases - can be found at:

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 additionally very often interleaved audio, timecode and metadata) in a single file on the storage. Autodesk Flame works with several different clip-based codecs, including:

- RED

<http://www.red.com/learn/red-101/redcode-file-format>

- XAVC

<https://en.wikipedia.org/wiki/XAVC>

- ProRes

https://en.wikipedia.org/wiki/Apple_ProRes

- DNX

https://en.wikipedia.org/wiki/Avid_DNxHD

<http://www.avid.com/products/avid-dnxhr-and-dnxhd>

Compression

Both image sequences and clip-based media may be compressed. Compression can yield benefits in required storage capacity and throughput but can generate computational overhead and reduce the perceived visual and audio quality of the media.

Compression may be lossless (where the exact original image data may be recovered), lossy (where some image data is discarded in the compression process), or 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, particularly visual effects artists, normally work with uncompressed image sequences. Consequently, it's important that architects design a full system that can support the high performance demands of these workflows.

Autodesk Flame media makes the highest demands of workstations, networking and storage in any implementation, and so 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.

There are five important factors that dictate the size (and consequent throughput requirements) of media files, and their impact on the demands of supporting systems:

- Frame Size
- Frame Rate
- Color Depth
- Encoding Method
- Compression Method

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

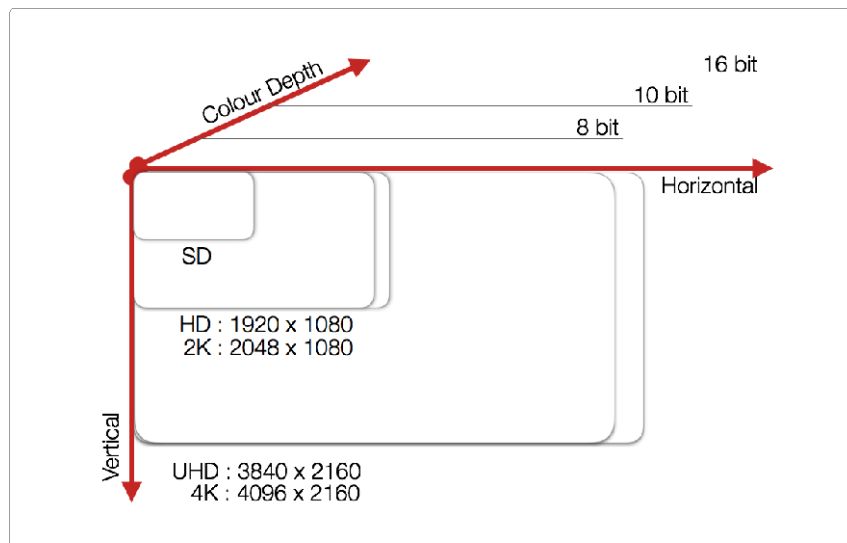


Figure 1 Relative frame size and color depth

As an example, listed below are the throughput requirements for 10-bit DPX sequences for several size and frame rates.

Name	Horizontal Size (Pixels)	Vertical Size (Pixels)	Frame Rate	Frame Byte Size (MBytes)	Throughput (MBytes per second)
HD	1920	1080	24 fps	7.91	190
2K (DCI)	2048	1080	24 fps	8.44	203
UHD	3840	2160	24 fps	31.64	759
4K (DCI)	4096	2160	24 fps	33.75	810
HD	1920	1080	60 fps	7.91	475
2K (DCI)	2048	1080	60 fps	8.44	506
UHD	3840	2160	60 fps	31.64	1898
4K (DCI)	4096	2160	60 fps	33.75	2025

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 after it is requested by the application.

For image sequence workflows (where each frame of video is stored as a separate file), it is assumed that the application normally will request each frame sequentially whilst the preceding frame is being displayed. It follows therefore 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 42ms for 24 fps material all the way down to 17ms for 60fps material.

1.4 Basic infrastructure guidelines

Workstation Specification

Dell 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, but more information can be found at:

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

Isilon and OneFS Version

Isilon F800 All Flash is qualified to support Autodesk Flame 2020 environments. OneFS version 8.1.2 has been tested and qualified and is required as a minimum. Later versions of OneFS may deliver equivalent or improved performance.

Network Connectivity

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

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

The choice of Network Interface Card, and recommendations regarding network interface optimization are dependent on the specific environment of each installation and are therefore outside of the scope of this document.

Filesharing Configuration

In a collaborative environment, multiple users, using multiple workstations are required to access and share files. The Autodesk Flame application is hosted by CentOS Version 7. Standard guidance when using any distribution of Linux is to use the NFS filesharing protocol. It is assumed that the most appropriate variant of the protocol for most environments will be NFSv3.

The creation of an NFS Export on the Isilon storage cluster can be achieved using the administrative web interface of OneFS. Details are available at:

<https://community.emc.com/docs/DOC-75133>

Care should be taken to architect a OneFS NFS export and permissions strategy that accommodates the required workflow. NFS optimisation guidelines for connections from CentOS are dependent on the specific environment of each installation.

Configuring OneFS NFS exports and designing a permissions strategy is outside of the scope of this document.

2 Isilon OneFS optimization guidelines

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

- Metadata Read/Write Acceleration
- Streaming and Concurrency Modes
- Filename Pre-Fetch

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 filesystem metadata, which can improve access time and reduce latency. Where all underlying storage units are SSD, as in the case of Isilon F800 All Flash, then this is not a meaningful or required optimization.

The qualification for Flame 2020 was performed using Isilon F800 All Flash, however previous version of Flame were also qualified against an Isilon H600 hybrid node with a mix of 10K SAS drives and SSDs. When using the H600 node with Flame, this optimization is relevant.

Enabling metadata read/write acceleration

Metadata read/write optimization can be enabled as a File Pool Policy at the OneFS Web User Interface. More information and guidance are available in published documentation:

http://doc.isilon.com/onefs/8.1.0/help/en-us/index.html#ifs_c_ssd_pools.html

Streaming mode

Streaming performance of OneFS can be optimized by enabling Streaming Mode at the file pool, or at the directory level. Streaming performance is important when playing media linearly - for example when playing a video sequence in a time line.

Consideration should be given to other workflows hosted in the same pool or directory before enabling Streaming mode. Best performance for all workflows on a single cluster might be achieved 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 (disc drives or SSDs).
- Data is prefetched aggressively.

Concurrency Mode

Due to streaming mode's aggressive prefetching of data, there are circumstances when multiple clients are playing back large video clips when Flame may start to drop frames due to excessive prefetching for a particular client. In those circumstances enabling Concurrency mode on the directory containing the media may be preferable.

In general, administrators should set media directories to streaming mode. If Flame is dropping frames when the storage should otherwise support playback, switching the media directories to Concurrency Mode is a relatively quick to change to try and observe if there is a positive (or negative!) impact.

Choosing streaming vs concurrency mode

Streaming mode is optimized for a small number of clients playing back high throughput material such as ProRes 4444 4K @ 60fps. 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 @ 30fps), 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 or concurrency mode

Streaming or Concurrency mode can be enabled as a File Pool Policy at the OneFS Web User Interface. More information and guidance are available in published documentation:

http://doc.isilon.com/onefs/8.1.0/help/en-us/index.html#ifs_t_configure_default_io_optimization_settings.html

http://doc.isilon.com/onefs/8.1.0/help/en-us/index.html#ifs_t_modify_file_and_directory_properties.html

It should be noted that when Filename Pre-Fetch (outlined below) is also required and enabled, there is an interaction between the two optimizations - which must be considered when configuring.

Filename Pre-Fetch

Filename Pre-Fetch can be enabled to optimize the streaming performance of OneFS when working with image sequences. It is critical that this optimization only be applied to directories containing image sequences of successively named files. Applying it to directories that do not contain image sequences may result in *false fetches* (where resources are used pre-fetching files that are not ever requested by the client).

When Flame is being used to work with image sequences, then improved performance may be achieved by enabling Filename based Pre-Fetch. Where image sequences are not used in the environment, Filename based Pre-Fetch should not be enabled.

Operation

Filename Pre-Fetch enables a performance optimization by detecting when image sequences are requested by a client. When a sequence is detected, OneFS will pre-fetch files from the underlying storage units before they are requested.

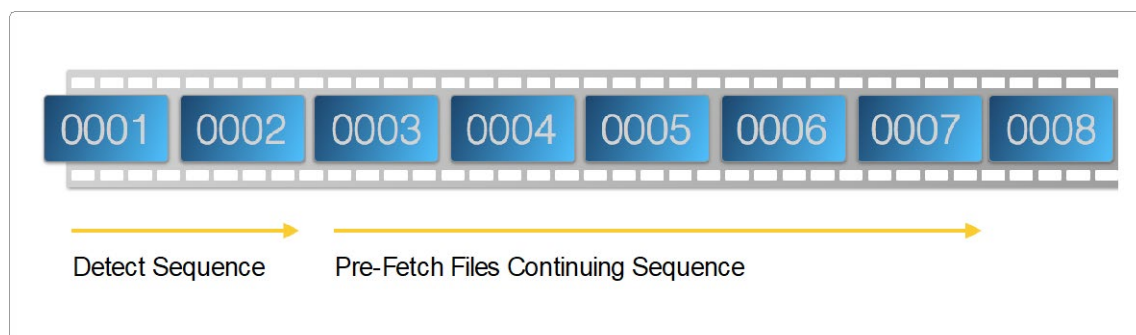


Figure 2 FileName Pre-fetch example

The FileName Pre-Fetch algorithm is optimized to detect image sequences with either decimal or hexadecimal filename numerical increments.

Enabling

Filename Pre-Fetch can be enabled on individual directories at the command line of OneFS. A detailed overview and configuration guide is available at:

<https://dellmcevents.com/uploads/Dell-ECM-Isilon-Filename-Based-Prefetch.pdf>

The Filename Based Pre-Fetch guide also provides good explanations of setting and examining OneFS file layout and access strategies using the command line interface and is recommended reading even if Filename Pre-Fetch is not to be used.

Further information

For detailed guidance on File Pool Policies and data layout and access strategies, check:

<https://www.emc.com/collateral/hardware/white-papers/h10719-isilon-onefs-technical-overview-wp.pdf>

Guidance for using the command line to make administrative changes is available at:

<https://www.emc.com/collateral/TechnicalDocument/docu84280.pdf>

Guidance for using the web interface to make administrative changes is available at:

<https://www.emc.com/collateral/TechnicalDocument/docu84277.pdf>

3 Autodesk Flame 2020 environment variable optimization

The `Environment Variables` noted in this document change the way that the application interacts with its host workstation. It may be tempting to increase certain settings to a very high level to try and achieve a maximum improvement. However, this may have a detrimental effect including the possibility causing application instability and crashes.

Any changes that are made should be tested thoroughly before deploying in a live environment. It is recommended that only the minimum deviation required from default values to achieve the required performance increase be configured. 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. Concurrency may be increased and therefore throughput performance improved by increasing the number of threads.

Threads for Managed Partition

The number of threads maintained by the application when reading from a *Managed Partition* may be set by using the `Environment Variable`:

```
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.

The number of threads maintained by the application when reading from an *Unmanaged Partition* may be set by using the `Environment Variable`:

```
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 ahead of their being required - and stage them in workstation RAM.

If more frames are pre-fetched by the application and stored in workstation RAM, the environment will better tolerate any transient latency issues, however they may be caused.

The number of frames that the application will attempt to keep in its buffer may be increased, and the amount of RAM allocated to the process may be increased. In testing, increasing the number of frames has been shown to have a positive effect on the latency and stability of application playback, resulting in significantly fewer dropped frames.

The default value is 40

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

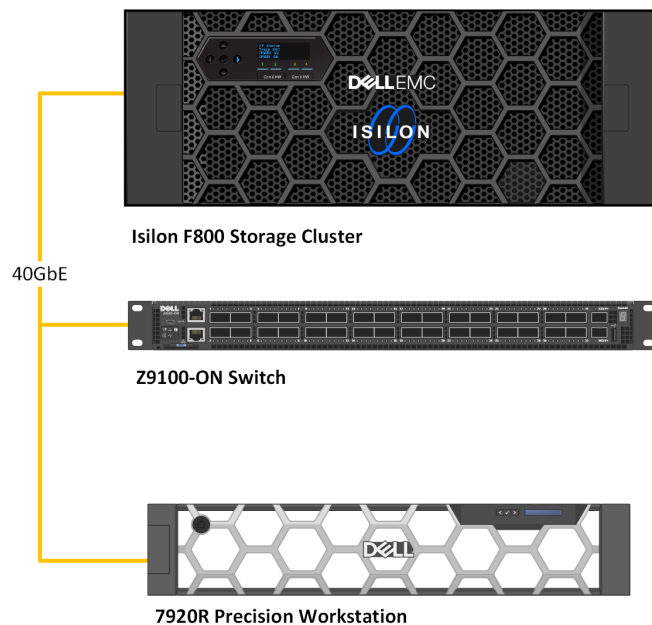
The number of frames may be increased by setting the Environment Variables:

```
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 resource (at the application, client workstation and storage) consumed by overly aggressive pre-fetching and the amount of workstation RAM available.

4 Architecture used for Flame 2020 storage certification

Dell Technologies & Autodesk Flame 2020



Management and back-end Isilon networking omitted.

Figure 3 Flame 2020 storage certification architecture

Isilon storage passed the Autodesk Flame 2020 Storage certification with an Isilon F800 storage array, Dell Z9100-ON switch, and Dell Precision 7920 rack-mounted workstation. Detailed system requirements for Flame 2020 can be found here:

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

Isilon F800 storage cluster

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

Dell Precision 7920R rack mount workstation:

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

Dell Z9100-ON network switch

- 40GbE Networking from Isilon cluster to workstation

A Technical support and resources

[Dell.com/support](https://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.