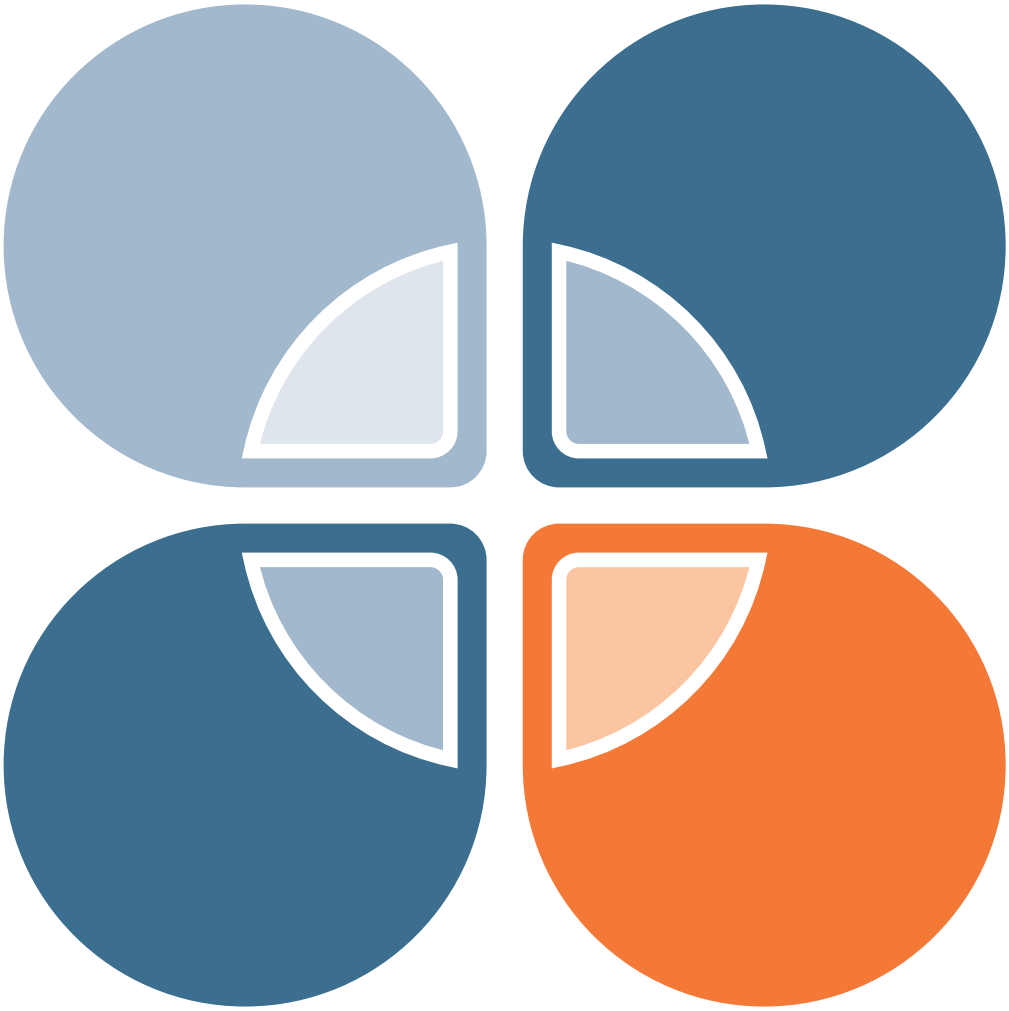
WHITE PAPER

EditShare XStream EFS Shared Storage

Managing increasing data demands and resource contention challenges in Media & Entertainment workflows



Introduction

Today's media production workflows are extremely demanding. High resolution files and large teams of collaborators push storage systems for greater amounts of bandwidth and larger, more efficient storage capacities. The EditShare XStream EFS Shared Storage System is a scale-out media storage solution that has been specifically designed to address the challenges of today's media production workflows. This new system is based on the EditShare File System, a parallel file system that uniquely addresses the needs of a modern media production organization by overcoming resource contention issues that normally impact the performance shared storage systems.

Key Components of the EditShare XStream EFS Storage System

In its simplest form, an XStream EFS Storage system is configured as shown below in Figure 1. It consists of a metadata controller (top left) and 3 storage nodes (top right). The metadata controller can be thought of as the traffic cop of the system. Client devices (NLEs) make read/write requests to the metadata controller and it responds with the locations to perform the read/write operations.

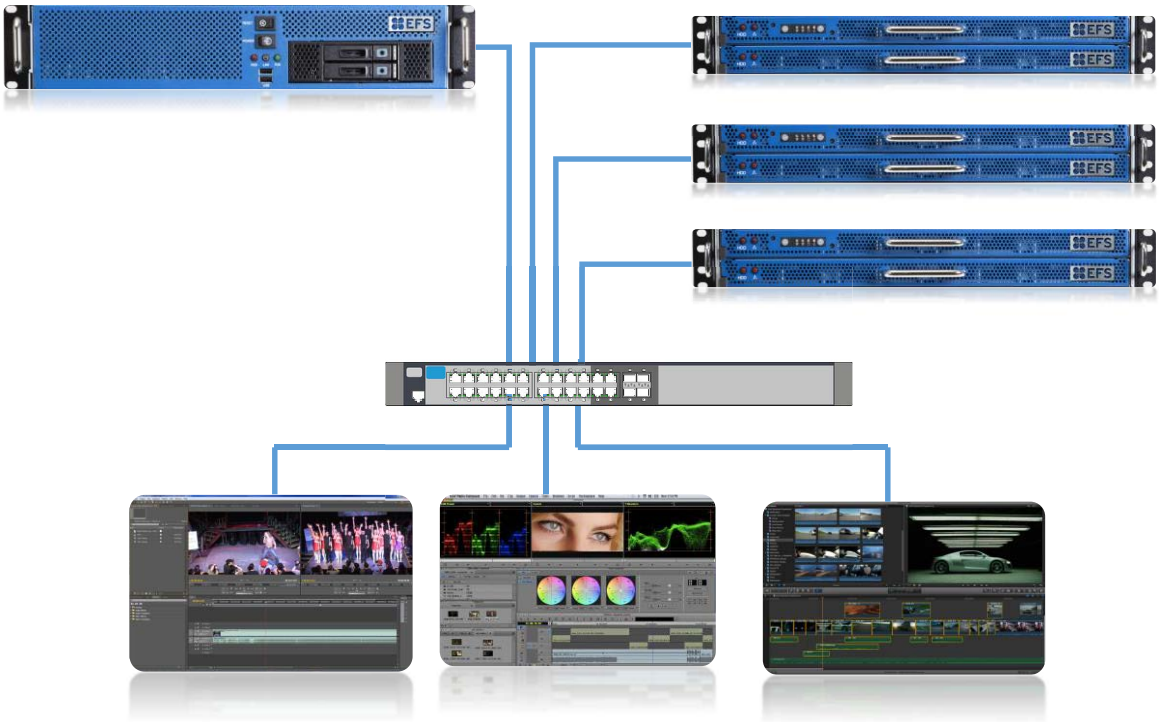


Figure 1 Example of an EditShare XStream EFS Storage System

Overview of EFS operation

To understand how the EditShare File System (EFS) manages the behavior of the storage system, we'll look at typical data write and data read operations. In most applications of the XStream EFS Shared Storage System, client devices are non-linear editing systems (NLEs) or other media processing workstations and each client device is equipped with driver software that enables it to communicate with the EditShare File System.
When a client device requests to write a file, the EditShare File System instructs the client to break the file into a number of data blocks, to calculate the parity data associated with those blocks, and to write the data blocks and parity, in parallel, to all of the storage nodes of the system. Thus, the designation of the EFS as a parallel file system.

A key performance benefit is the fact that, when distributed in this manner, the resulting bandwidth of the write operation is the sum of the bandwidth available from all the storage elements (i.e. all disk drives) in the system.

The number of data blocks is determined by the XOR[1] configuration of the storage system. The system illustrated in Figure 1 has an XOR 2 configuration; data blocks will be written to 2 of the nodes and the associated parity will be written to the remaining node[2]. In the general case, the EFS always reserves one node for parity[3]. Thus an EFS system will have an XOR (nodes -1) configuration. There is no limit to the number of storage nodes in an XStream EFS Shared Storage System.

Reducing Metadata Latency

As previously described, client devices communicate with the metadata controller to obtain location information and permissions to write data to or read data from storage nodes. It is important for the metadata controller to respond to read/write requests as quickly as possible; any delays caused by the metadata controller can directly impact read/write processes. In some commercial storage systems, metadata is stored on hard disk drives. The result of this approach will be that metadata operations will be delayed by the average latency of the disk drives used; typical enterprise drives have latency figures ranging from 50 – 150 msec. Similarly, some commercial storage systems combine metadata management and other system functions, such as storage, in the same hardware. Contention for hardware resources is possible with such an approach.

Recognizing the importance of fast metadata operations, EditShare XStream EFS Shared Storage System provides a dedicated metadata controller whose role is to respond to metadata transactions as rapidly as possible. In addition, unlike some other storage solutions, in the EditShare XStream EFS metadata controller, the metadata itself is stored in fast RAM to avoid hard disk latency.

Reading Stored Data

When a client device requests to read a file from storage it, again, communicates with the EditShare File System and is given the information it needs to locate the stored data associated with the requested file. The Client retrieves those blocks and reassembles the file for use in the client application.

The XStream EFS Shared Storage System is designed for media production environments where multiple editors share project spaces and media assets. Typically, each editor uses an NLE (client). It is a natural consequence of the environment that due to factors such as concurrent requests for the same data, the nature of the write process or simply bad luck, as more media files are distributed across the storage nodes and as requests to read files are made, contention for access to the storage nodes will increase. And in this environment, contention is unacceptable because of the potential to slow down or delay access to a given file and the potential to interrupt processes like live playout or cause video or audio frames to be dropped when content is being rendered.

[1] XOR is a Boolean function

[2] This is a simplification to aid in understanding of the behavior of the system. In reality, data and parity are written across all nodes. A complete description, therefore, is “data blocks will be written to storage space equivalent to 2 nodes and the associated parity will be written to storage space equivalent to 1 node.

[3] This is the same simplification. The EFS always reserves storage space equivalent to one node for parity.

EditShare *Swift Read* Technology

To deal with the issue of read contention, the EditShare File System monitors the performance of the storage nodes and detects which ones are responding more slowly and may delay read requests. Armed with this critical information, it can instruct a client device to read all of the data blocks associated with the requested file or, if a node containing data is experiencing contention and is likely to delay a read request, to skip that node and reconstruct the requested data using the parity information instead. By reconstructing instead of waiting for a slow resource, data can be retrieved at a faster rate than is normally possible in the presence of contention for storage resources.

An added benefit of the EditShare File System is that the same performance monitor that detects slow responses from a storage node can also detect when a storage node has completely stopped working, as in the case of a hardware failure. Thus the XStream EFS Shared Storage System is able to continue operation even if an entire storage node discontinues operation and, therefore, delivers an unprecedented level of fault tolerance.

Efficient Use of Storage

The final benefit of the EditShare File System is the efficient way in which it makes use of storage resources and how that efficiency increases as system size increases. In the case of the example in Figure 1, two storage nodes are used for data storage and one node is used for parity. Thus, the XOR 2 configuration illustrated uses 2/3 of its storage resources for data and 1/3 for parity. In an XOR 3 configuration, 3 nodes (3/4 of storage resources) are used for data and 1 is used for parity. An XOR 4 configuration uses 4/5 nodes for data, XOR 5 uses 5/6 nodes for data and so on.

Other high performance shared storage systems simply make duplicate copies of data to avoid contention and improve read speeds. This approach, known as mirroring, can at best use just 50% of storage resources.

Summary

As we have shown, the EditShare File System is a parallel file system that uniquely addresses the challenges of the modern media production environment. The EditShare File System enables the XStream EFS Shared Storage System to provide the performance necessary to handle the bandwidth demands that accompany HD, 2K, 4K and beyond high definition formats. This system provides a collaboration-friendly workspace for teams of creators working on feature film and television content. And it provides a highly-reliable, highly-available, fault-tolerant repository for the media assets so necessary to sustain the media and entertainment marketplace.