PROXY WORKFLOWS FOR A SECURE REMOTE PRODUCTION FUTURE

Stephen Tallamy

EditShare, UK

ABSTRACT

Using lower resolution proxy files for video production has been an option for a number of years, however advancements in technology, remote working demands, and a drive towards remote production and usage of cloud technologies has opened up a range of new possibilities.

Modern proxy workflows depend on a set of underlying technologies to provide a seamless, secure, and productive user experience. The paper will identify enablers for remote production and provide concrete examples to illustrate how media professionals:

- Overcome challenges of collaborating with high resolution footage
- Control access to valuable original footage
- Create unique renditions to simplify workflows and increase traceability

INTRODUCTION

For many years, post-production teams have enjoyed the benefits of dedicated editing suites with access to high throughput local networks attached to central storage. Working within a controlled environment allows teams to work directly with the source media in the highest possible quality throughout the production process.

As various countries of the world went into lockdown due to the 2020 pandemic, an immediate realisation was that access to high resolution footage would be limited. A production using high/ultra-high-definition formats presents a significant challenge for both the total amount of storage required and the bandwidths required to stream video to a remote editing workstation.

Lower resolution proxy files have been used in various use cases: for example, where an editing workstation is not powerful enough to process media at higher resolution, where the original material is currently in offline storage, or where the client application (for example a web browser) does not support the format of the original material.

As an illustrative example, consider a one-hour documentary filmed in UltraHD (3840x2160) at a frame rate of 23.98p and captured in Apple ProRes 4444 (132.6 MB/s) format. If we assume a 100:1 shooting ratio (i.e., 100 hours are captured for the one-hour documentary) then the total storage required (ignoring any storage redundancy) is 45.5 TB and a four-layer sequence requires a streaming bandwidth of 132.5 MB/s. In many cases this exceeds the amount of storage you would have available in a remote location and requires significantly higher bandwidth to stream the originals than would be available from typical internet connections. Using an h.264 encoded proxy file at a lower resolution can yield usable proxies at around 800 Kbit/s bitrate, bringing the total storage required down

to 34 GB, well within the reach of an internet download or single portable drive. Table 1 shows a range of illustrations for other example productions using different codecs.

Use Case	Resolution	Codec	Frame Rate	MB/sec	Hi-res total file size (TB)	Proxy total file size (TB)
1-hr Nature Doc 100:1 shooting ratio	3840x2160	ProRes 4444	23.98p	132.5	45.49	0.034
Non-Fiction TV 1000 hours	3840x2160	XAVC 300	23.98p	30	103.00	0.343
	1920 x 1080	XDCAM 50	59.94i	6.25	21.46	0.343
	3840x2160	ProRes 422 HQ	59.94p	221	758.74	0.343
2-hr Feature Film 200:1 shooting ratio	6144 x 3456	ARRI RAW	24p	732	1005.25	0.137
	7680 x 4320	RED RAW 5:1	24p	260	357.06	0.137

Table 1: Storage requirements for different types of production

While proxy workflows have been available for many years, the shift towards remote production and cloud enabled workflows, plus some recent advances in technology and computing power, enable new, useful opportunities for proxy workflows across a wide range of tools.

CORE TECHNOLOGIES FOR PROXY WORKFLOWS

A key component of any proxy workflow is, naturally, a transcoder capable of converting the original media from its raw format into a usable proxy. While many transcoders are oriented around delivery formats (MPEG2/MPEG4) for both input and output, an encoder for proxies must be capable of reading and creating video in formats such as DNxHR, ProRes, XAVC and a range of RAW formats from companies such as Red, Blackmagic, Sony, Canon and ARRI.

Many of the popular non-linear editing (NLE) applications provide embedded transcoding capabilities to facilitate the creation of proxy files. A number of cameras and capture devices also have built-in encoding capabilities that can create proxy files in parallel to the original high-resolution format.

A popular workflow, however, is to use a centralized encoding capability that can be triggered through automated services that scan for new media added to storage spaces or other triggers (for example the source material passing some initial automated quality check). This workflow provides a lot of flexibility to encode media from a wide range of input sources and removes the need to consume edit workstation time with encoding jobs.

Centralized encoders can be dedicated hardware solutions, embedded software encoders (such as MainConcept and ffmpeg) or cloud services. Cloud services such as AWS Elemental have added support for a range of production formats over recent years, enabling a very scalable encoding capability (provided the high-resolution media can be transferred to storage within the cloud). Figure 1 shows some examples of professional

output formats from AWS Elemental MediaConvert that can be utilized by an open proxy workflow solution.
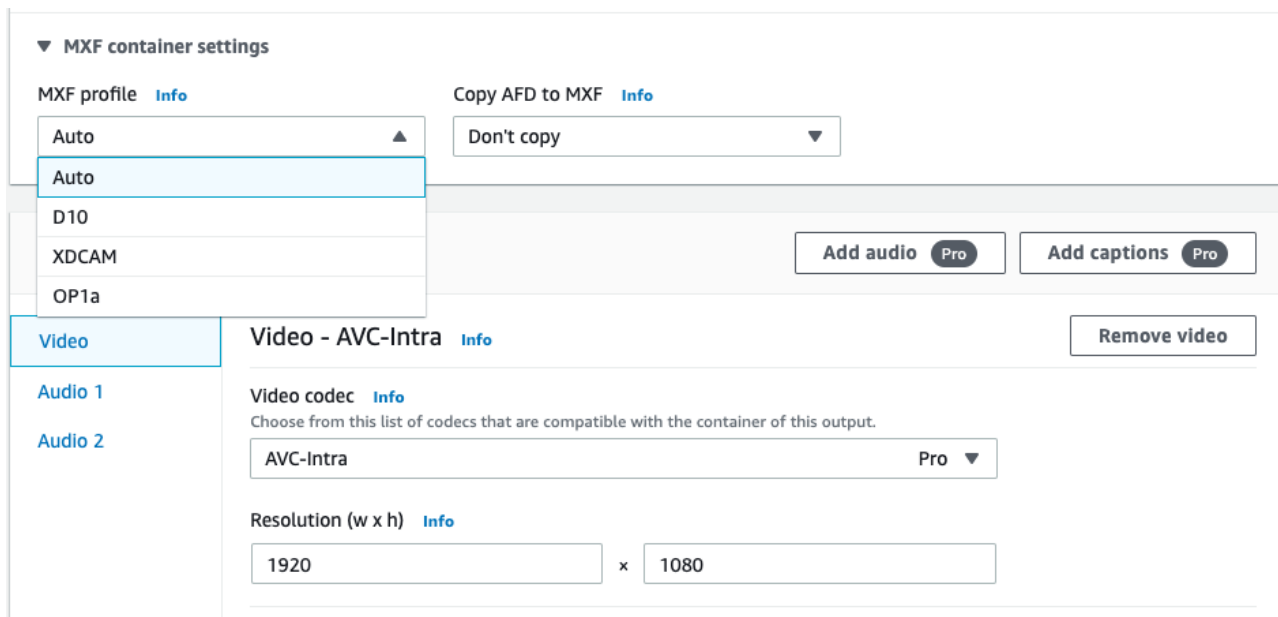

Figure 1: AWS Elemental MediaConvert production format support

In some cases, a blend of techniques may be used to best utilize the various components of a solution and optimize for certain use cases; for example, to enable an edit-while-capture workflow with proxies or to batch process a large archive of material. It may also be necessary to generate multiple proxy formats to cater to different client software, for example an mp4 format for a web client and a QuickTime format for an editing client.

To provide this flexibility, another key component will be an automation and job management system. This system will need the flexibility to execute rules depending on the input format and proxy usage. For example, in a production that uses a blend of HD, UltraHD and 8K source formats, different options may be required for the proxies to ensure the resolution/bitrate is appropriate still for editing, as shown in Figure 2.
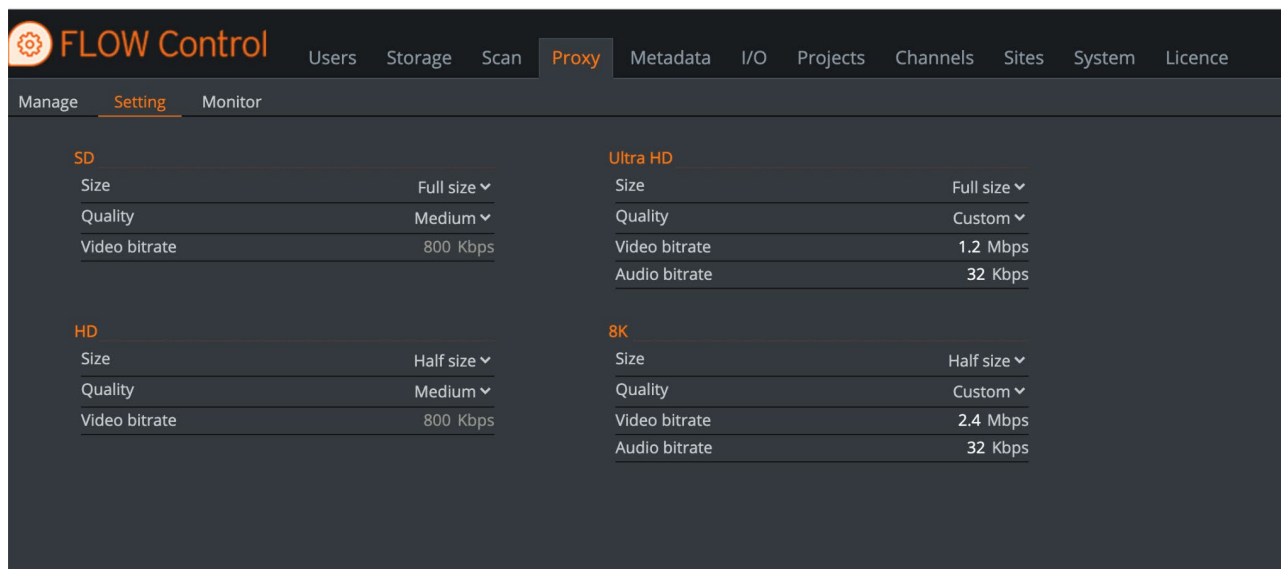

Figure 2: EditShare FLOW proxy format configuration options

Job management is also a critical part of the system, for example to allow high priority footage to be processed into proxy format ahead of other footage or to distribute work across different encoders (with different capabilities and throughput). Figure 3 shows the performance benefits of utilizing compute capacity across multiple available servers when combined with appropriate job management to maximize throughput.
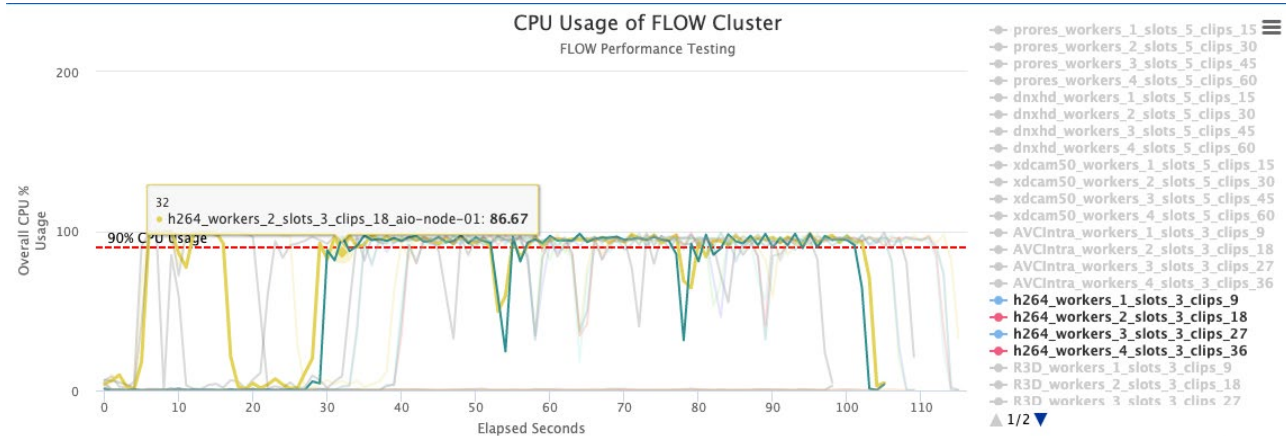

Figure 3: Proxy load distribution across multiple worker nodes

To utilise these proxies, a flexible storage platform is required to support the storage of both the high-resolution material and the proxy files. Clients will need to access these proxies through a range of different protocols, with the primary methods being via a local network file mount and HTTPS streaming/download. To address the widest range of client applications a local storage client provides ultimate flexibility as it can expose different storage backends as a single coherent mount point that can be used exactly the same as any storage locally attached to the workstation. The storage client can use the appropriate protocols to access storage in multiple storage tiers and locations without requiring custom plugins for the client application. Figure 4 shows the EditShare Connect storage client that is available for Windows, macOS and Linux, allowing users to select from a wide range of storage spaces for local mounting.
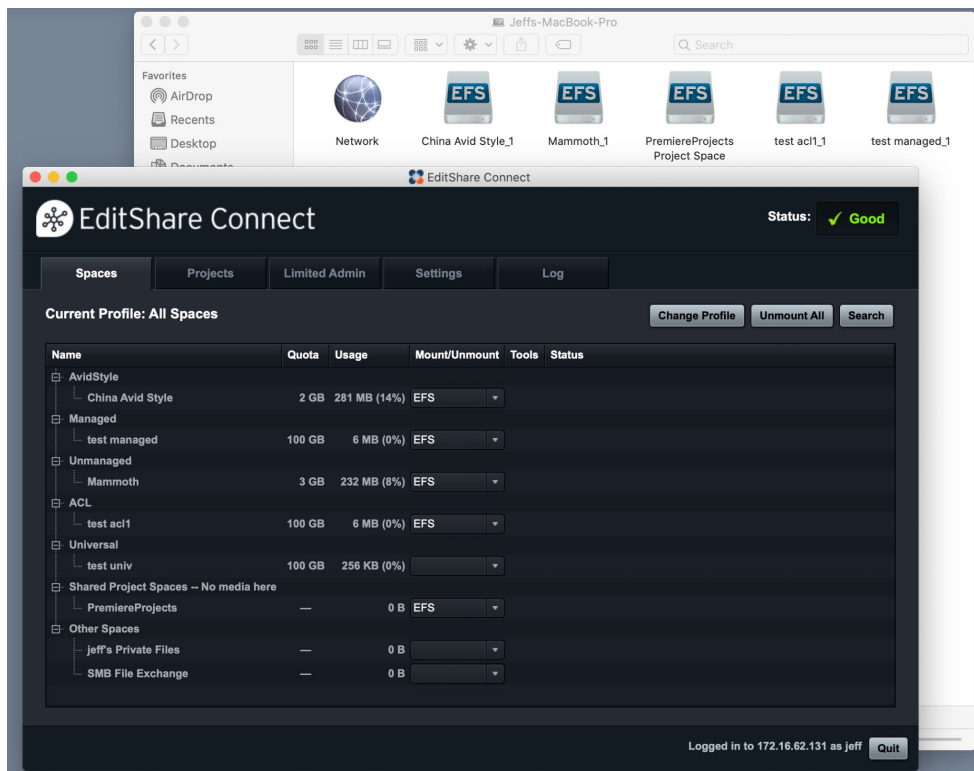
Figure 4: EditShare Connect with selected storage spaces mounted on macOS

The final component is functionality within the client application to toggle easily between the proxy resolution of any asset and the original material for rendering/conforming as well as for reviewing high resolution to ensure that any video artifacts in the compressed proxy are not present in the source.

Many of the predominant non-linear editing applications have the ability to relink a timeline sequence to a different set of files representing an alternative encoding of the source material. This relinking can automatically match files by a range of attributes including filename and metadata within the file (for example MobIDs, original tape, etc). This is often used to create a duplicate sequence within the same project and to protect the original sequence from any potential corruption that can happen if the relinking operation picks up the wrong media or fails to find an appropriate file for linking. Figure 5 shows an example of the relinking options with Avid Media Composer.
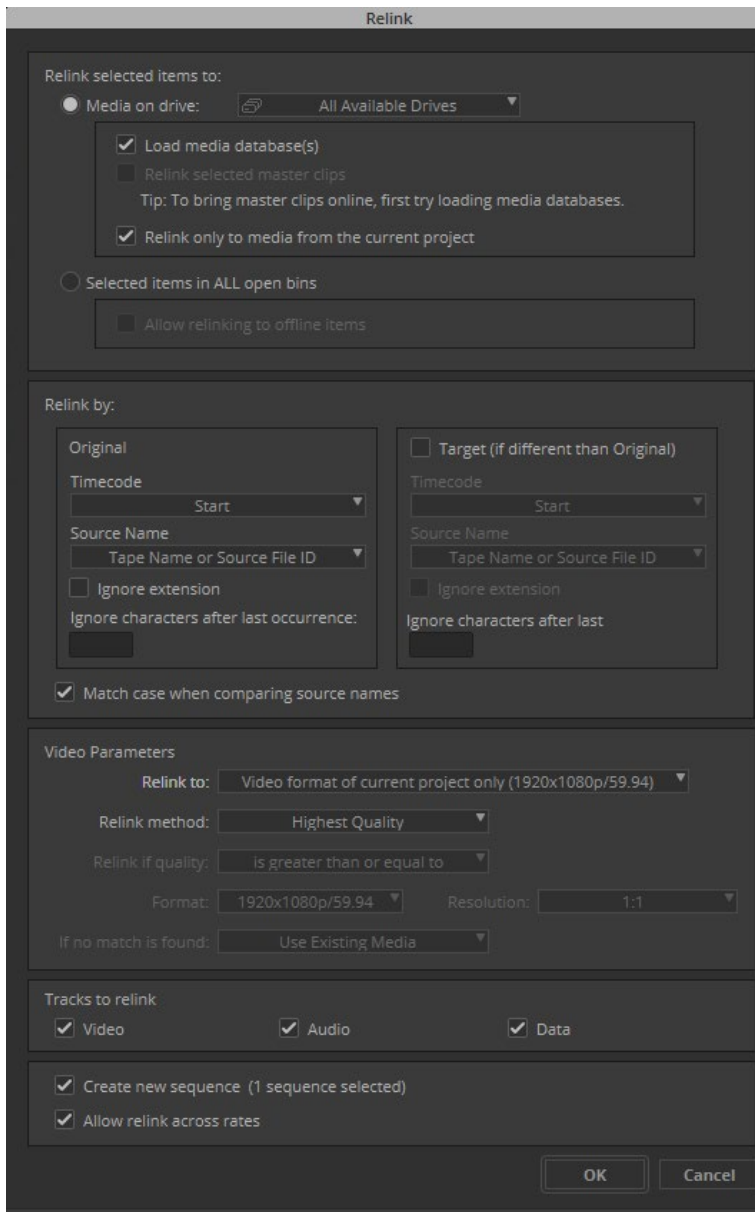
Figure 5: Relink options within Avid Media Composer

More recently, some non-linear editing tools have supported the ability to link any imported media clip to multiple file locations, thus allowing users to toggle between proxy and high resolution both in the source monitor and also within a full sequence. Figure 6 shows this functionality for Adobe Premiere Pro and Figure 7 shows it within DaVinci Resolve.
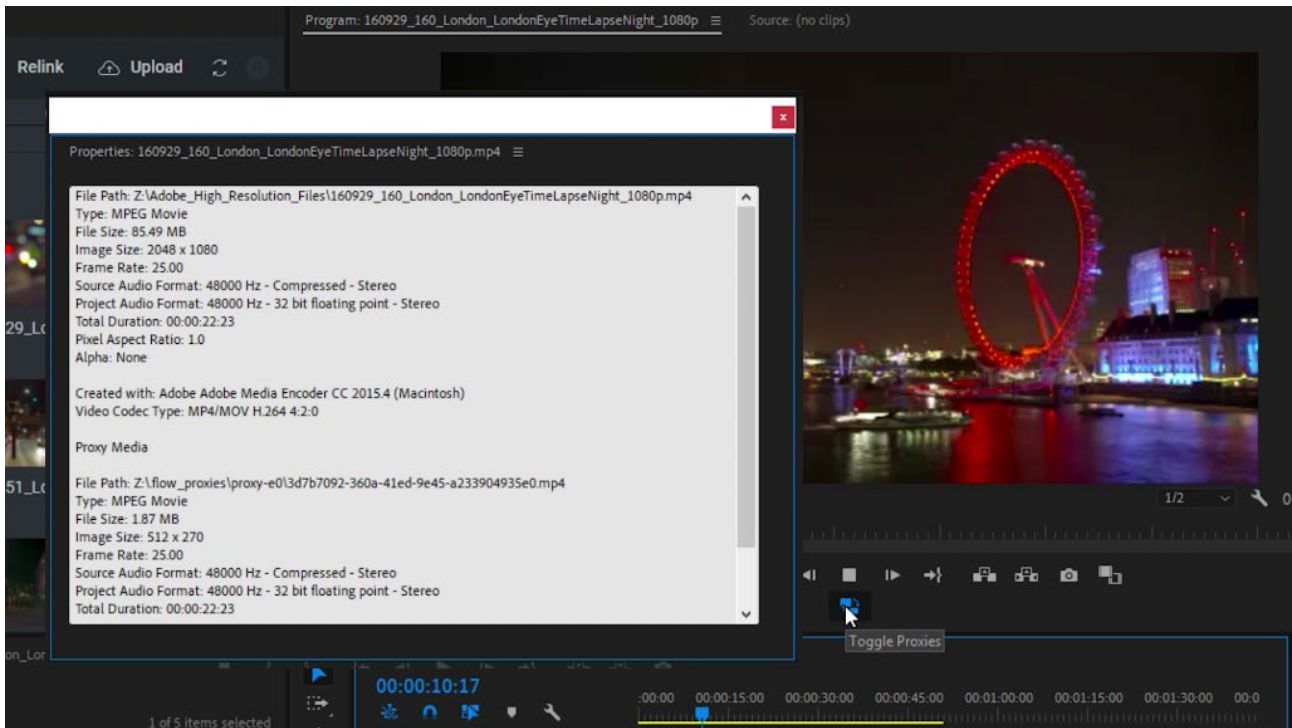
Figure 6: Adobe Premiere clip properties showing high resolution and proxy resolution that can be switched between using "toggle proxies" button
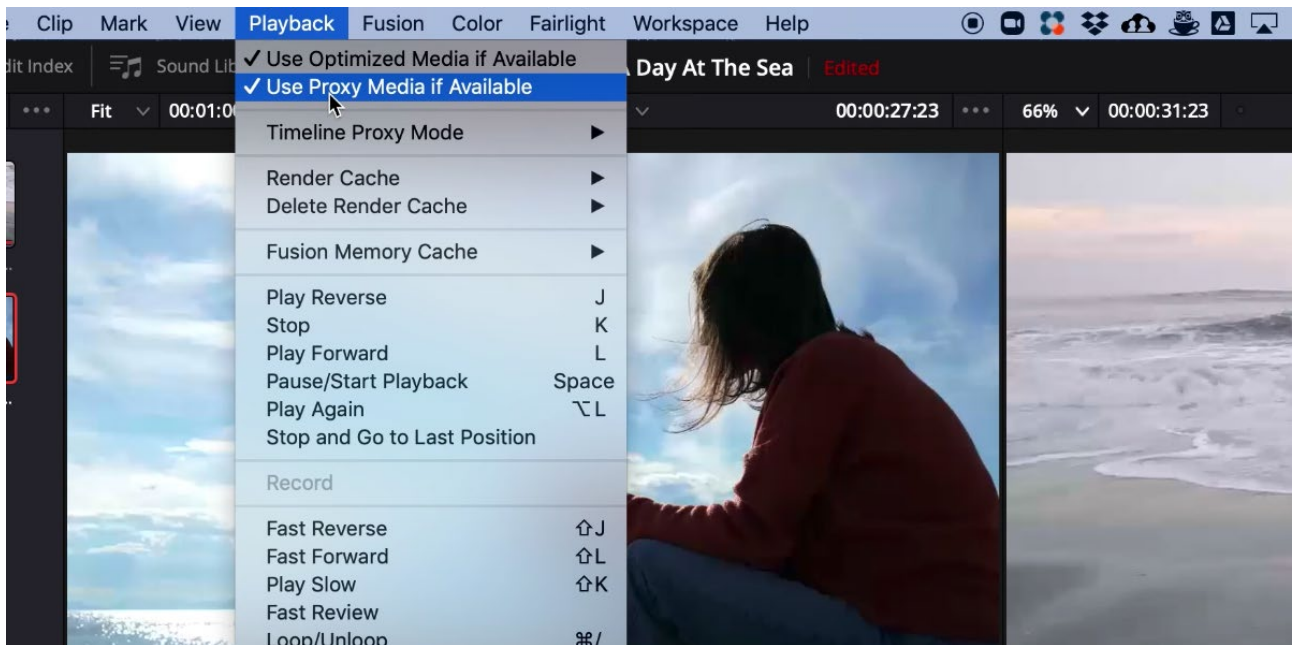

Figure 7: DaVinci Resolve playback options to use proxy media

For those tools that do not natively support this capability, external tools can be used to either perform the same functions via APIs into the editing application (if the tool provides Open APIs) or alternatively by exporting via a common format, such as AAF, processing that file format, and then re-importing it into the tool. These more complex workflows often add complexity to the production process, leading many creatives to prefer to avoid proxy workflows completely. Removing friction from the storytelling process is key to adoption of any technology solution.

CLOUD STORAGE COST OPTIMISATION

With the core components of a proxy editing system at hand, the next steps are to reduce the friction through automated workflows and seamless integration to hand off between the tools. This can enable a wide range of collaboration experiences that work even when users are in remote locations. They can also be used to facilitate cost optimizations that can take advantage of different storage tiers, including cloud native storage such as Amazon Simple Storage Service (S3).

The appeal of cloud native storage such as Amazon S3 is obvious, it provides a highly resilient store at a much lower cost than virtual disk storage such as Amazon Elastic Block Store (EBS).

As an example, consider the 45.5 TB of storage required for the documentary production illustrated earlier, with prices from AWS as of April 2021 in the region us-east-1. This production would incur a raw storage cost in Amazon EBS of $2,620, when including resilience across multiple servers in an XOR4 redundancy pattern. Using Amazon S3 the standard storage tier would cost just $1,071 / month with the ability to reduce this price to $582 / month for infrequently accessed media. Table 2 uses the previous production use cases to outline some of the storage costs assuming XOR4 across clustered servers is used for EBS storage protection.

Use Case	Resolution	Codec	Frame Rate	EBS Cost / Month	S3 Standard Tier Cost / Month	S3 Infrequent Access Tier Cost / Month
1-hr Nature Doc 100:1 shooting ratio	3840x2160	ProRes 4444	23.98p	$2,620	$1,071	$582
Non-Fiction TV 1000 hours	3840x2160	XAVC 300	23.98p	$5,933	$2,372	$1,318
	1920 x 1080	XDCAM 50	59.94i	$1,236	$505	$275
	3840x2160	ProRes 422 HQ	59.94p	$43,704	$16,879	$9,712
2-hr Feature Film 200:1 shooting ratio	6144 x 3456	ARRI RAW	24p	$57,902	$22,180	$12,867
	7680 x 4320	RED RAW 5:1	24p	$20,566	$8,095	$4,570

In addition to the raw storage costs, there is no need to pre-provision storage within Amazon S3, it grows as you need it. Attached storage such as Amazon EBS tends to be allocated in blocks and you pay for the full block, regardless of usage, meaning the savings could be much higher, particularly at the start of a production.

The other appeal of cloud native storage such as Amazon S3 is that it integrates well with other cloud native services such Machine Learning and Artificial Intelligence, which make use of it's simple RESTful API and role-based security.

One issue with S3 storage is it is not a POSIX compliant file system and so it lacks key features often used by editing applications such as file ownership and symlinks. It cannot be mounted directly as a file system and so workflows involving S3 either require movement of files between S3 and a mountable filesystem or some form of gateway or client application to be usable in many tools.

Proxy editing optimized storage management technology (figure 8. EditShare EFS) solves this challenge by separating the POSIX metadata from the storage of the raw files. This means that EFS can blend storage that is served from an EBS backed storage node or from an S3 bucket. The storage client (figure 8. EditShare Connect) can present the structure of the filesystem natively to the operating system using FUSE and other native libraries. When a file is requested for read, the client will request the backend location and then natively stream the file using the storage management native protocol or the S3 protocol.

The storage management client authenticates via the metadata controller, allowing administrators to share access to a range of different storage spaces that may be backed by any combination of EBS storage nodes or S3 buckets. The authentication can be against a user store within the storage management system or external such as Active Directory. The storage management metadata controller can then provide appropriate authentications to the client to access S3 buckets, meaning the user does not require an AWS Identity and Access Manager (IAM) user or API keys.
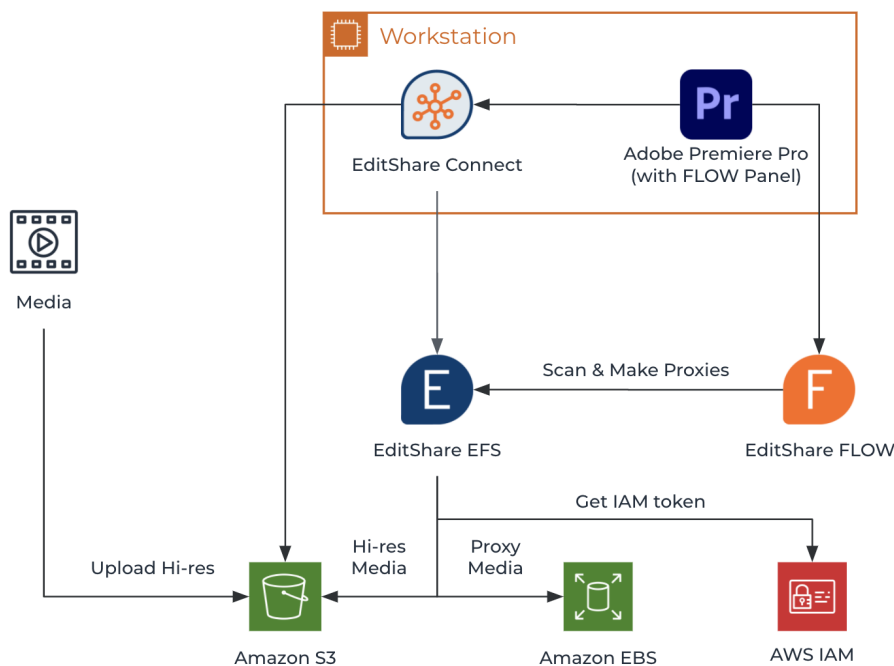

Figure 8: Cloud editing architecture with blended EBS and S3 storage

The next challenge of S3 is the latency of read access. Whilst S3 has excellent throughput it can exhibit variable latency which is usually overcome in linear video playout systems by prefetching the next segment of video once a linear stream has been established. This means the latency is only really experienced at the start of the linear stream and variability can be evened out through prefetching.

Non-linear editing, however, presents challenges because it is much less predictable where the next video segment will be fetched as, during the editing process, the user will need the ability to quickly jump between different parts of a timeline and so any lag will create an unusable experience. The entire contents of an S3 bucket could be downloaded directly to a workstation, however this then pushes up the storage cost on the workstation, negating much of the benefit of using S3 over EBS. For offline rendering it presents less of a challenge and so streaming directly from S3 can work really well for conforming sequences, but not so well for the editing process.

This is where a proxy workflow can overcome the drawbacks of S3 by providing a lower bitrate video on a different, low latency, storage location for editing and then providing seamless access to the high resolution for rendering, where latency is not a concern. With both proxies and high resolution mountable with a transparent storage client that blends the backend storage and automation to generate the proxies it is possible to create a seamless proxy workflow that can dramatically reduce infrastructure costs in the cloud.

An additional benefit of working with proxy media is that a virtual workstation in the cloud can be modestly specified for editing tasks, with only render workstations requiring higher CPU and memory allocations for conforming. This can provide substantial cost savings.

SEAMLESS PROXY WORKFLOW

As mentioned previously, key to the success of a proxy workflow is how the technologies blend without introducing friction into the process. The following provides a sample cloud-based proxy workflow that enables the cost optimizations described above without creating overhead in the process.
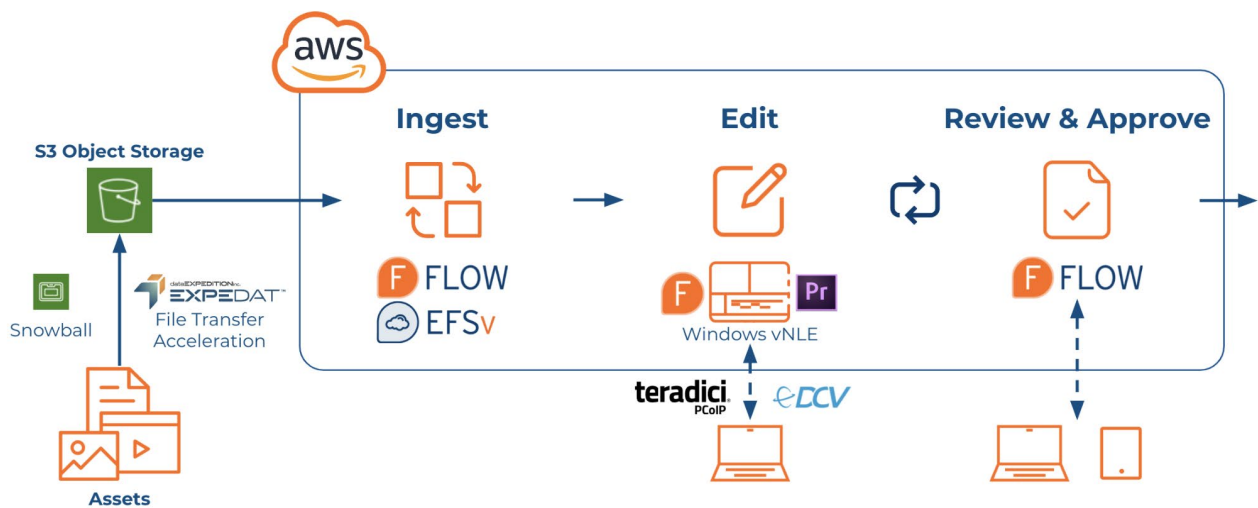

Figure 9: Seamless Proxy Workflow for EditShare cloud solutions

The original material is uploaded to a standard S3 bucket in AWS using whatever mechanism is appropriate. This could be via a standard S3 sync tool, using an accelerated transfer, such as CloudDat from Data Expedition, via the AWS console, or even via an AWS Snowball device if large volumes of media need to be imported.

Within the seamless proxy editing storage management tools, a mediaspace is created that links the S3 bucket to a virtual storage space within the seamless proxy editing storage filesystem. A mediaspace can be used as a collaboration space for multiple users to access a subset of storage available in the system. As the material is added to the S3 bucket, the storage management metadata controller is notified of added/updated/deleted files and maps these into the appropriate mediaspace. It appends additional metadata for the files based on the default user permissions and other settings within the mediaspace.

In addition to mapping the file location into the metadata, the solution also automatically scans the files to create assets within the integrated media management database and generate proxies. These proxies are generated by the media management application which directly streams the high-resolution video from S3 using the storage management Linux client. The media management application writes the encoded files back to storage

spaces within storage management that are persisted on low latency EBS volumes. This will ensure that clients reading the proxy media will always have a low latency experience.

The NLE virtual workstation, deployed as an EC2 instance, can directly attach to the storage management solution via the connect desktop software and browse available storage spaces. It can select a mediaspace backed by S3 and browse the files as it would any other network mount.

End users access the virtual workstation via a remote desktop technology such as Teradici PCoIP or NiceDCV. These remote desktop technologies are designed to provide ultra-low latency and high-quality graphics that are required for creative tasks such as video editing. They make use of GPUs to stream the desktop experience over an internet connection across multiple displays and provide remote keyboard/mouse controls and audio. The result is almost indistinguishable from a local desktop experience. To enable Teradici a GPU enabled virtual machine is required such as the AWS g3 and g4dn EC2 instance types. These instance types embed Nvidia graphics cards which are used by both Teradici and the video editing software.

Users can then create a project in the NLE and use the media management Panel capabilities to select media to add to the project. The panel can display the proxies to allow the user to preview content before adding it to the project and see any other associated metadata such as markers or predefined subclips.

The media management application is tracking the multiple locations of the file and presents the user with the option to import both high resolution and proxy files into the NLE as shown in Figure 10. These become linked within the same clip inside the NLE. The file paths are automatically mapped to the mounted location as selected by the connect application, removing any possibility of user error. This then allows the user to use the Toggle Proxies option within the NLE.
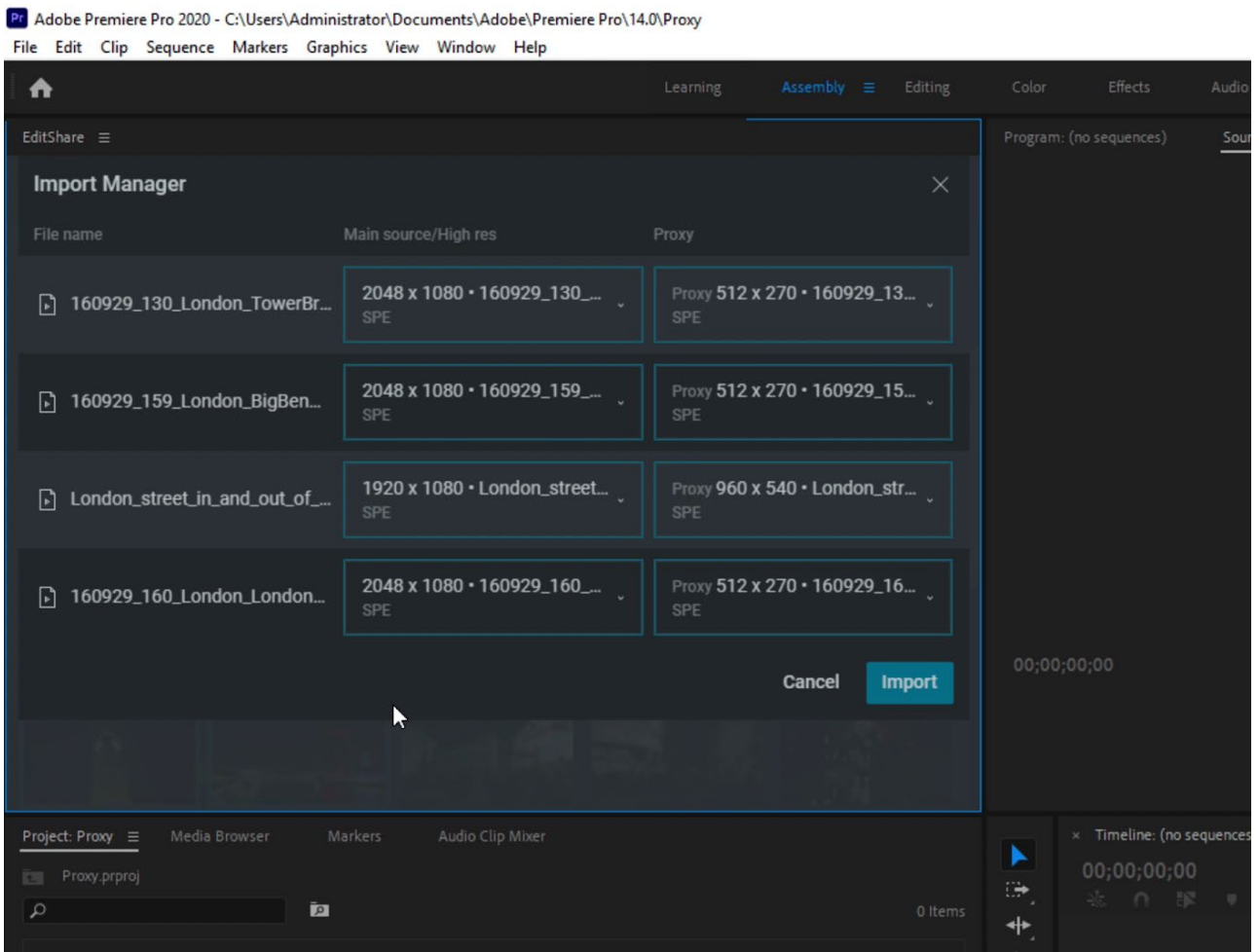
Figure 10: Import options within EditShare FLOW Panel for Adobe Premiere Pro

The user can then build up a full sequence timeline with effects using the proxy media, which is streamed with low latency. Should the editor need to view more detail from a particular clip they can toggle to the high resolution, with the knowledge that there may be some slight latency while the clip loads.

On completion of an edit the user can switch back the high resolution and use the panel to render the sequence and upload the result back to the storage and asset manager for review. This rendering can happen on the same workstation or the project can be loaded on another workstation dedicated to conforming.

While this particular workflow is partly designed to overcome current latency limitations of AWS S3, which may eventually be solved by architectural changes to S3 itself, the workflow still keeps projects lightweight during edit and these techniques can be used to optimize for other benefits outlined below. Also, this workflow is not limited to usage in the cloud, editors working remotely can make use of this workflow to reduce the bandwidth required to access centrally stored material and even work offline when required.

EXTENDED USE CASES

Having selected compatible, extensible and open components and built a seamless workflow that allows for multiple representations of the same media, a range of extended

use cases become possible. For example, with high value content it can be important to restrict access to the raw footage and track media usage from those accessing material.

One workflow example use case would be to disable access to the original material to users who would be required to work in proxy format. The proxies can be varied per user or group of users, for example by burning unique watermarks/fingerprints into the proxy media for security purposes. With a detailed understanding of production lifecycle and media usage, these proxies can be generated either on-demand or just in time, to avoid wasted encoding time and storage. Within a cloud-based environment these options become even more viable with massively scalable infrastructure for encoding and storage.

All of this infrastructure can be combined with detailed access auditing that monitors file usage throughout disparate storage backends, providing organizations with accountability and triaging tools to understand who accesses which file, when, and where. When fed into machine learning tools, this data can also be used to detect anomalous access patterns and alert operations teams with the ability to prevent access to media that may be compromised. Tools such as the EditShare EFS Auditing solution provide an open approach to gather detailed file system access logs that are viable within built-in ElasticSearch / Kibana but also exportable to external systems for additional analysis. By utilizing the metadata controller architecture, EFS also can track usage of files across a range of backends in combination with the EditShare Connect storage client.

CONCLUSION

While initially considered for offline editing purposes, the proxy workflow continues to find new applications for remote production and cloud. Key to creating a seamless proxy workflow is a set of open technologies that can be adapted to different use cases such that the experience for editors and other creatives is transparent and frictionless. Coupled with embracing cloud native storage and advanced services such as machine learning and artificial intelligence, this combination can unlock new ways of collaborating and reaching talent across the globe.

REFERENCES

1.   AWS Pricing Calculator: https://calculator.aws