Technical Guidelines for Final Cut Pro

Best Practices for Final Cut Pro and Isilon IQ Storage Platform with OneFS 6.0

An Isilon® Systems Technical Whitepaper

December 2010
Version 2.0

Table of Contents

Introduction

Final Cut Pro (FCP) is a professional non-linear editing software application developed by Apple Inc. It runs exclusively on the Apple Macintosh hardware platform, including MacPro and iMac desktops and MacBook Pro portable systems. Released in 2000, FCP has the ability to edit many digital formats including DV, SD, HD, 2K and other MXF and QuickTime-compatible codec formats. According to 2009 and 2010 Apple sales data, FCP unit sales represent more than 50% of the professional editing market, with Avid at 24% and a variety of other edit applications from Adobe, Sony, Grass Valley and others making up the remainder of units sold. Final Cut Pro has a devoted userbase and developer community thanks to the adoption of the QuickTime engine for its ease of use and 3rd party plug-ins.

In small-scale deployments, Final Cut Pro captures video to the Mac's local hard drive or RAID-based direct attached storage, where it can be edited and rendered. An FCP editor can mix clips using different codecs from multiple sequences and edit them on a single timeline. As the FCP project's size and complexity grows, it becomes extremely difficult to manage and share all this content. In larger Final Cut Pro editing environments, shared-editing storage can be provided using SAN-based technology such as Apple Xsan, but such storage systems are difficult to manage, scale, and share with other applications and platforms, resulting in additional islands of storage and IT data management complexities.

Unlike traditional SAN storage and RAID array systems, Isilon IQ scale-out storage offers a far simpler high-performance shared-editing storage system for Final Cut Pro workflows. With the Isilon scale-out platform, multiple editors can share content, working on different stages of individual projects, or on multiple projects simultaneously. Editors can use different post-production applications for transcoding, editing, rendering, and delivery, while an IT administrator can manage, tier and backup a single pool of storage.

Several factors have occurred to drive the rapid adoption of Isilon scale-out storage as a shared-edit infrastructure for Final Cut Pro editing workflows:

1. Mac OSX 10.6 Snow Leopard with an improved NFS network stack has made NAS storage a far more viable option for file-based media workflows as opposed to complex and costly Fibre Channel and Ethernet SAN infrastructure.
2. The Isilon IQ X-series scale-out storage platform with the Accelerator-X and OneFS 6.0 software have brought extraordinary advances in the ability to support mixed SD, HD, 2K and RED workflows.
3. Isilon has further established itself as a leading storage provider for Final Cut Pro workflows by partnering with numerous media vendors including (but not limited to) Harris, Front Porch Digital, Telestream, Gallery SIENNA and Aspera for WAN media distribution.

This document describes the benefits of using Isilon IQ scale-out storage as a single volume platform for a wide array of Final Cut Pro workflows. It also covers the environments supported and configuration steps necessary to achieve those performance results. Readers are assumed to have knowledge of both the basic architecture and features of Isilon IQ scale-out storage as well as Final Cut Pro and Mac OS X.

Storage Challenges in Final Cut Pro Editing Workflows

When defining storage requirements for Final Cut Pro applications it is essential to understand common storage challenges facing editors and storage administrators.

High Performance and Storage Capacity

When HD video format support was introduced, direct-attached multi-disk storage arrays were sufficient to provide the performance and storage capacity requirements for most Final Cut Pro workflows. Edited content is later moved to a central repository from which other editors and artists continue to work and packaged content is transferred to its final point of distribution. While this may be acceptable in lower-resolution and small editing projects, it is impractical in the multi-editor HD world. Consider a project using the common DV50 video format that requires 7MB/s write performance during capture. The capacity for 1 hour of DV50 is 27GB. Compare this to a FCP project capturing video format of uncompressed HD 1080i 29.97 10-bit at 166MB/s write throughput. This HD capture requires a high-performance storage system capable of sustaining the data rate as well as 600GB storage for one hour of HD content. Now scale this to a typical feature production with 2.5 million feet of film, or 450 hours – this correlates to 270TB of storage per feature film. Digital file-based formats such as RED, Arri and Phantom create additional complexity for determining performance and capacity needs. In a multi-editor FCP environment, the difficulty of maintaining such performance load and capacity becomes unmanageable for traditional mixed local-attached and shared-storage strategies.

Scaling Storage Capacity and Concurrent Streams

At the center of the high performance and capacity challenge is the issue of scalability. Due to the high rate of data growth and dynamically changing performance requirements in today's post-production facilities, the average time between acquiring additional storage is only 90 days. But scaling capacity with traditional RAID-based SAN systems is complicated, time-consuming and often unreliable: RAID groups and new disk LUNs have to be added or existing LUNs may have to be reformatted or reconfigured, and there is storage unavailability during the expansion. Typical SAN configurations include a meta-data server that becomes the bottleneck despite the added storage hardware. As a result, scaling the storage infrastructure to support additional editing workloads or capacity needs typically requires creating separate storage systems that adds data management complexity, resulting in multiple storage silos.

Mixed Workloads

In most editing environments a mix of workloads occur concurrently on a shared project or different independent projects. Tasks such as ingest, edit and playout must support real-time data rates while other tasks such as rendering and transcoding do not require real-time streaming. Some tasks are more meta-data intensive such as when deleting an entire project comprised of thousands of files, while others are more throughput-intensive. In addition to sizing the overall performance requirements of a storage system, administrators must understand the different storage capabilities for read, write and meta-data operations, and ensure performance is met for each individual task by properly allocating and scheduling storage resources.

Dropped Frames

One of the main performance challenges encountered when using Final Cut Pro is the occurrence of dropped frames when capturing and playing video clips. This happens when

frames are not being streamed at the target frame rate based on the video format. If a frame is dropped during capture or playback operations, an entire sequence of footage must be re-captured or played back all over again. The reasons for dropped frames are numerous, for Final Cut Pro and other non-linear editing tools. Any component in the processing workflow that introduces data latency may result in dropped frames. These factors include insufficient client memory or processing power, inadequate video graphics card, slow internal bus links, slow links to storage, or slow storage I/O. Dropped frames are a frequent topic of discussion in many video editing forums and mailing lists, and Apple has published Final Cut Pro guidelines on how to address these issues in the following URL:

http://support.apple.com/kb/TS1165

Isilon IQ Scale-out Storage for Final Cut Pro Workflows

The introduction of 10-Gigabit Ethernet support in the Isilon IQ X-Series Storage Nodes and Accelerator-X platform, combined with OneFS 6.0, marks a significant leap forward to meet the growing demand for scale-out storage for Final Cut Pro and many other media workflow applications.

The Accelerator-X platform is a storage-free node with 10-Gigabit Ethernet network interface, (2) quad-core Intel CPUs and up to 32GB RAM. The Accelerator-X node is attached to the Isilon IQ cluster running the latest OneFS 6.0 distributed file system on top of X-series storage nodes each equipped with a quad-core Intel CPU, 4GB RAM and 12 disk drives of varying disk density. The Isilon IQ scale-out storage system was designed to address today's media workflow application challenges, and to deliver the following benefits:

1. The Accelerator-X facilitates up to 500MB/s single stream throughput via NFS for Mac OSX 10.6 clients
2. The X-series 2RU node provides (2) 1GbE interfaces, each fulfilling up to 100MB/s via NFS for Mac OSX 10.6 clients, or an optional configuration offering 200MB/s via 10GbE.
3. The X-series 4RU node and S-series 2RU node provides (4) 1GbE interfaces, each fulfilling up to 100MB/s via NFS for Mac OSX 10.6 clients, or 250MB/s via 10GbE.
4. New options for directory-based file layout for optimized streaming application read/write performance.
5. Multi-core CPU, SMP-enabled nodes in a distributed file system, and a customized pre-fetching algorithm to eliminate latency limitations during read operations.
6. Scale-out storage architecture to easily accommodate dynamic performance and capacity requirements (such as quantity of streams, and higher stream data rates).
7. Enhanced Flexnet network-management software to manage the Isilon IQ storage within a complex network topology.
8. The ability to create Data-Rate Shaping workflows, by combining Isilon's SmartConnect Advanced technology with pre-architected client device associations, to isolate Quality of Service bandwidth for specific devices.

Isilon Direct-Edit and Indirect Repository Workflows for Final Cut Pro

Isilon IQ can address performance requirement for two different Final Cut Pro modes of operations: High-Performance Scratch-Disk Storage, and Repository Storage. With the advent of the new SmartPools feature in OneFS 6.0, Isilon facilitates the ability to mix these functional and business requirements in one storage pool.

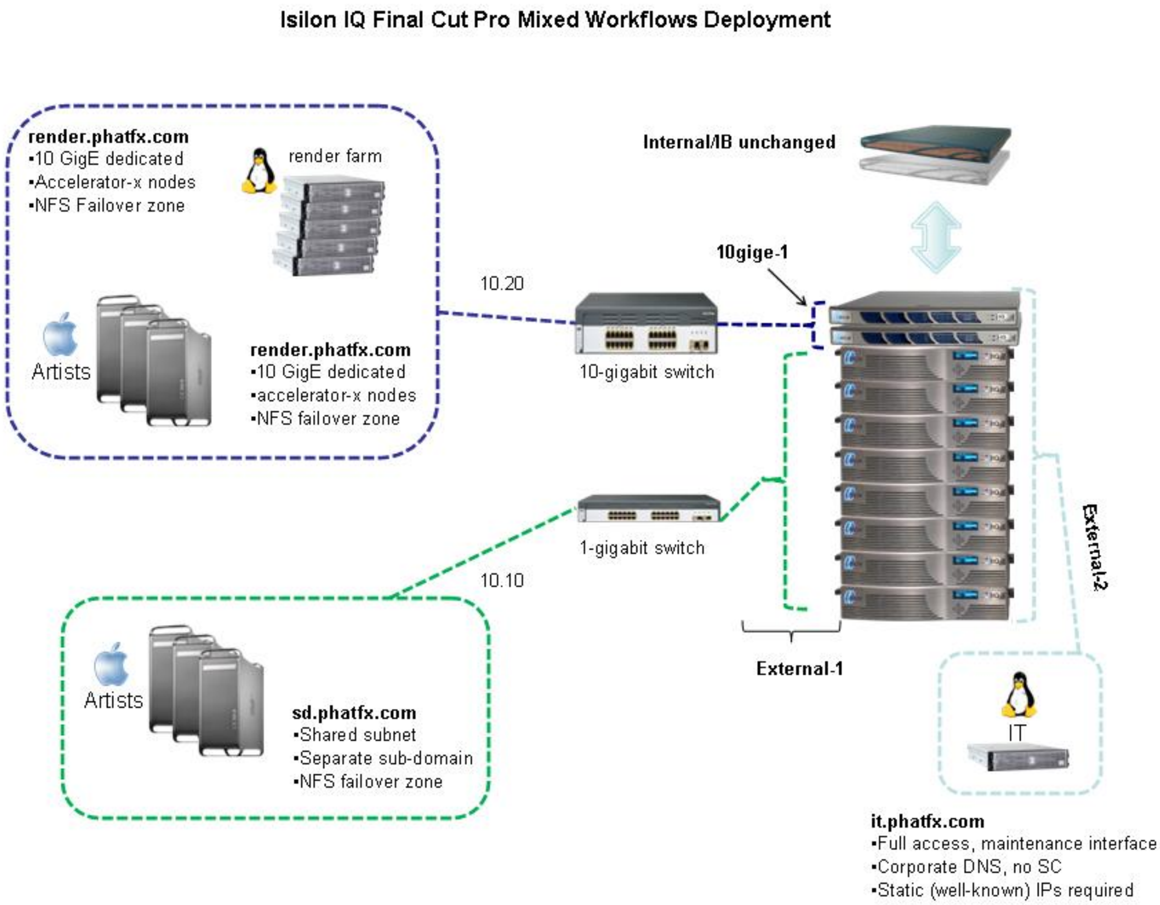


Figure 1: An example of the Isilon IQ scale-out platform deployed in a FCP mixed workflow using Isilon SmartConnect Advanced to create multiple connectivity subnets based on a direct-edit high-performance layer (sd.phatfx.com) and a second lower-performance layer (render.phatfx.com).

Using Isilon IQ as High-Performance Scratch Disk Storage

An Isilon IQ cluster can be used as a direct-edit workspace (FCP calls this a Scratch Disk) shared among many Final Cut Pro editors. In this real-time mode, media is captured, rendered or transcoded (written) to Isilon and played back (read) from the cluster. This allows multiple editors and artists to work on the same or different projects at the same time without the need to copy media across disparate storage silos. As more FCP workstations are added to the post-production facility, more nodes can be added to provide additional performance and capacity, without downtime. With OneFS 6.0, a large number of concurrent streams in compressed HD formats (such as Apple ProRes HD HQ) can be distributed amongst the Isilon storage nodes via the 1-Gigabit Ethernet interfaces. For uncompressed HD capture and editing, two options are available – the Accelerator-X diskless node can be added to an Isilon cluster providing 10-Gigabit Ethernet connectivity for the highest-bandwidth uncompressed video formats, and certain X-series and S-series storage nodes offer 10-Gigabit Ethernet as an option for mid-range bandwidth requirements.

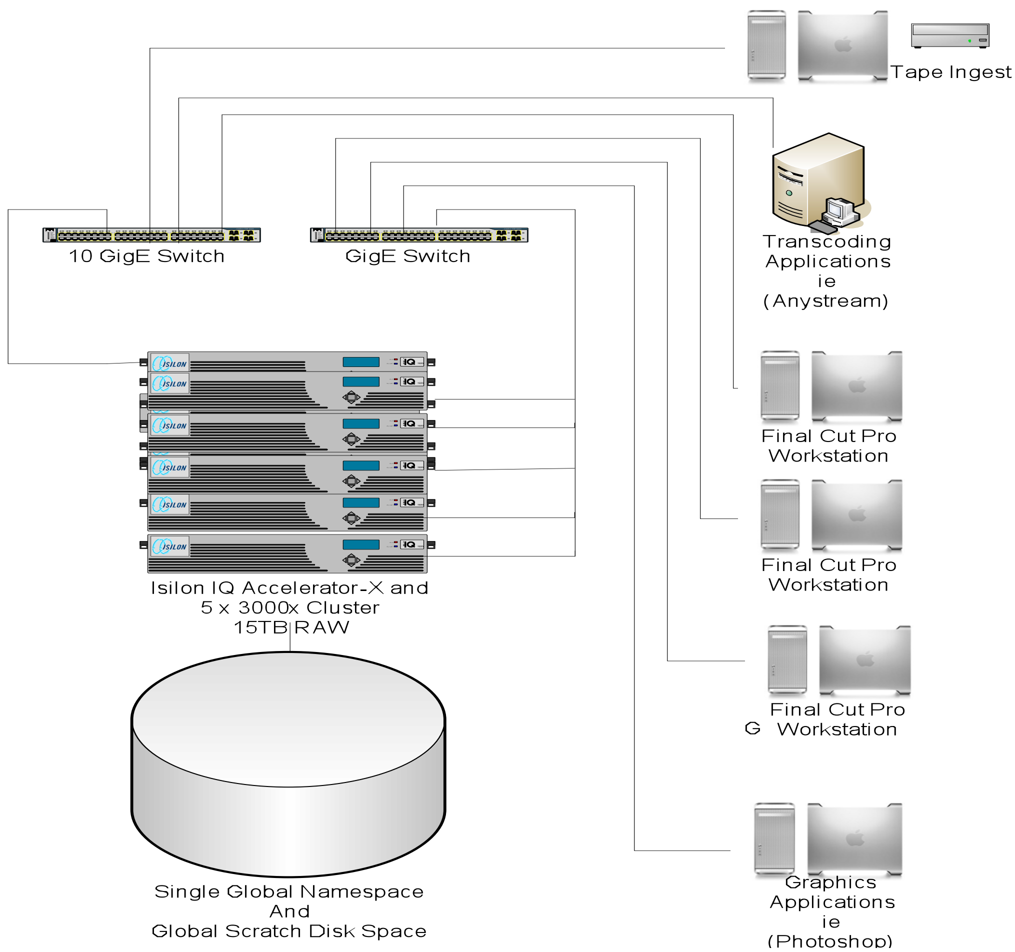


Figure 2: An example of a 5-node iQ6000x Isilon cluster with 30TB raw storage for a single direct-edit shared scratch disk space. The cluster includes 1 Accelerator-X node connected to 10-Gigabit Ethernet network. While some Final Cut Pro Macs route to Isilon storage nodes for compressed HD video processing (ProRes) via the 1-Gigabit Ethernet network, other FCP Macs connect to the 10-Gigabit Ethernet network for uncompressed HD capture and edit workflows.

Using Isilon IQ as Repository Storage

Isilon IQ can be used as a storage repository from which Final Cut Pro editors push and pull content to their primary local scratch disk space. In this mode, all content is consolidated, managed and protected within the Isilon IQ cluster, while FCP video capture and processing is conducted on the Mac's local storage. Since dropped frames are not a concern when moving data to/from the Isilon NAS volume, any type of content (such as DV/SD, HD, 2K/4K DPX or EXR, RED, and other formats) can be stored and protected, and large number of clients can connect simultaneously. Global bandwidth amongst all connected clients is limited only by the aggregate throughput of the entire Isilon cluster.

Using Isilon IQ as a repository storage system does not require any further application-specific performance tuning considerations. Repository storage can also be used as nearline storage for workflows such as rendering, color correction, or transcoding that do not require real-time video bandwidth but can still benefit from a high-performance scalable storage system.

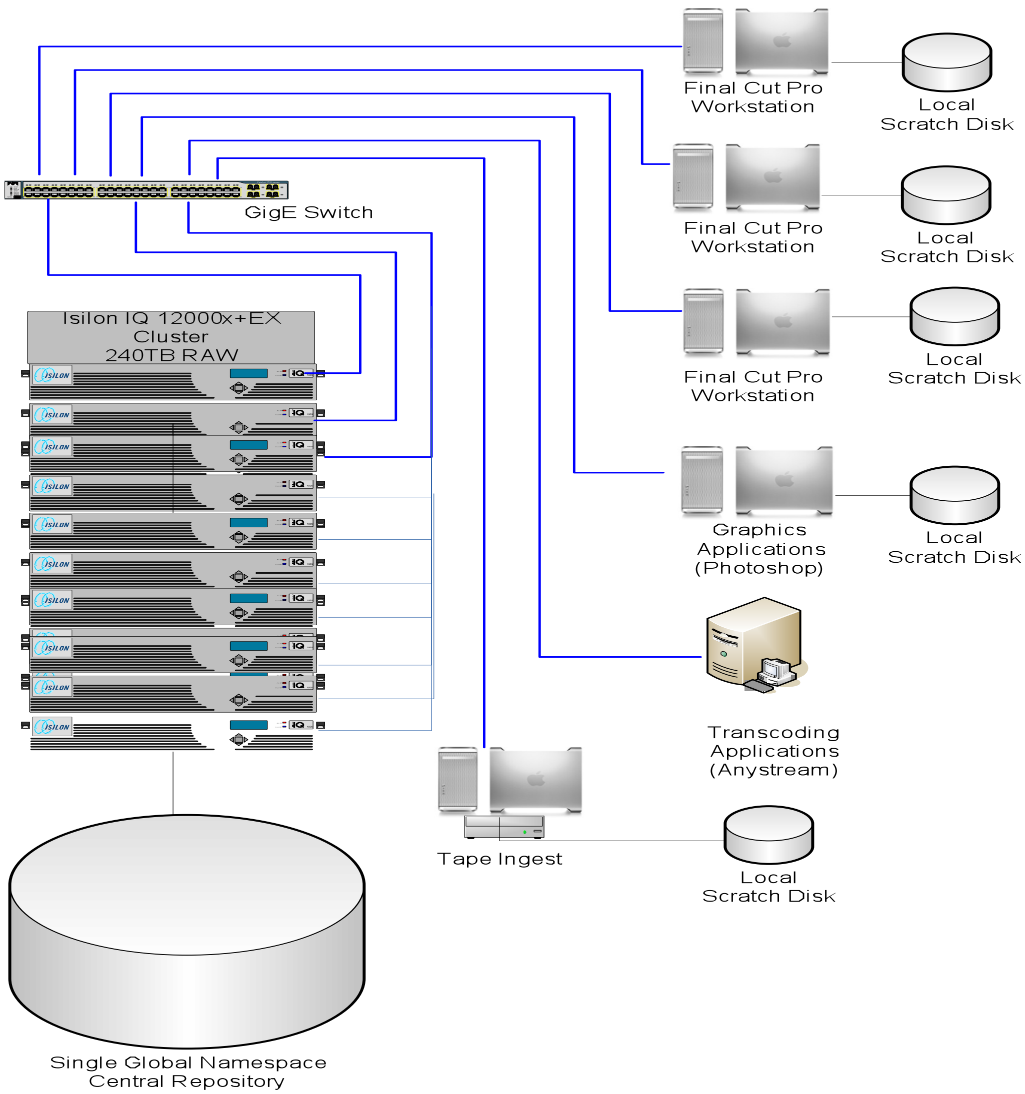


Figure 3: An example of an Isilon scale-out cluster with (5) iQ12000X + (5) EX nodes providing a total of 240TB raw capacity for storing and protecting Final Cut Pro content captured and edited locally on each Mac. This workflow also supports other media workflow applications such as transcoding and ingest.

Supported Final Cut Pro Formats Using Isilon IQ Scale-out Storage

To understand the range of supported video editing formats by Isilon IQ and leverage the benefits of scale-out storage, Isilon conducted a series of lab tests using Isilon IQ storage as a scratch disk space for Final Cut Pro Ingest, edit and playback workflows.

Isilon IQ Storage for Final Cut Pro SD/HD Workflows

The table below (Figure 4) lists various QuickTime SD and HD formats for FCP, and the tested stream count (per Isilon node). Two Mac FCP workstations were connected over a 1-Gigabit Ethernet network to a 5-node Isilon iQ6000X cluster running OneFS 6.0. Each Mac was connected to the same 6000X node.

Video Format	Throughput per Stream	Ingest Streams per Node	Edit Streams per Node	Playout Streams per Node
DV25	3.6MB/s	40	44	44
DV50	7.0MB/s	22	24	24
DV100 (DVCProHD)	13.7MB/s	10	10	10
SD 8bit	20.0MB/s	8	8	8
SD 10bit	27.0MB/s	6	6	6
ProRes HQ 1080i	30.9MB/s	6	6	6

Figure 4

The chart below (Figure 5) displays Isilon CPU utilization for the streams presented above – for each Isilon storage node, and across the entire cluster. Notice the red vertical dashed line delineating between the throughput load of an individual node (left side of line), and aggregate throughput via multiple nodes in the cluster (right side of line).

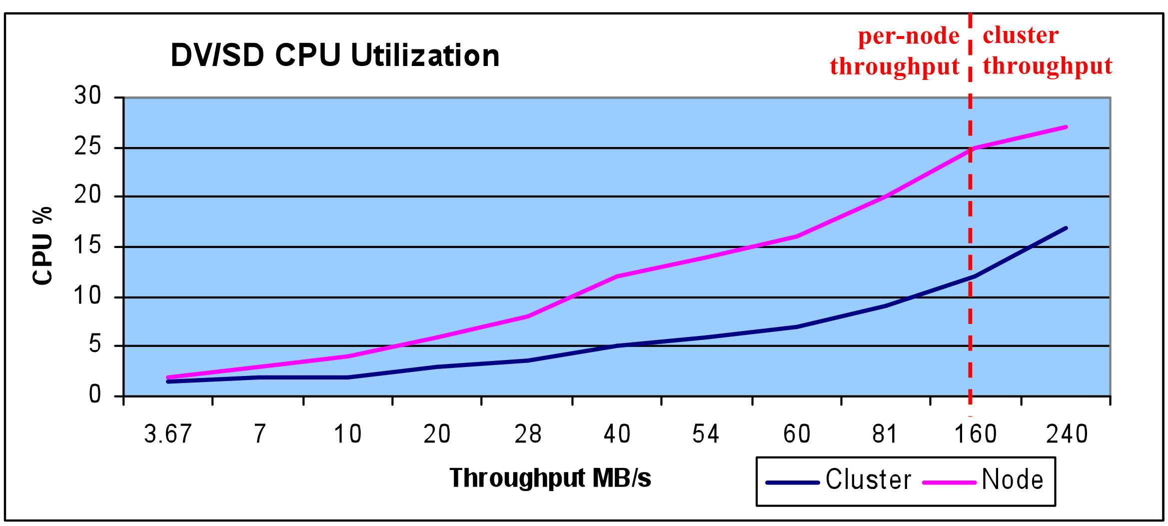


Figure 5

Sizing the Isilon IQ Storage Cluster in Final Cut Pro SD Workloads

Figure 5 on the previous page indicates the following for 1-Gigabit Ethernet workloads:

1. The throughput limit per storage node is 160MB/s for R/W. Latency overhead from disk management is the reason why throughput cannot maximize native NFS transfer speed across both 1-Gigabit Ethernet ports per 2RU X-series node.
2. As additional load is added on each node, the CPU usage increases nearly linearly.
3. As cluster aggregate load increases, cluster CPU usage increases nearly linearly.
4. As additional nodes are added to increase aggregate throughput, individual node CPU usage increases moderately.

Together, these metrics provide a simple way to project the number nodes required to support a stream count per video format and workflow. The linear scalability behavior in bandwidth and CPU usage shows that by adding more storage nodes for bandwidth disk I/O, more streams can be delivered by a single Isilon IQ cluster.

To size the number of storage nodes use the table in Figure 5 and do the following:

1. Divide the number of necessary video streams by the number of supported streams per node and workflow and calculate the total number of nodes necessary within a cluster.
2. Determine the density-type of storage nodes by dividing the total capacity by the number of nodes calculated in step one. For example, if 5 nodes are necessary to provide 800MB/s aggregate bandwidth and 24TB are storage capacity is desired, the design calls for (5) iQ6000X storage nodes.

Isilon IQ Storage and Accelerator-X Nodes

The Accelerator-X node powered by OneFS 6.0 provides 1-Gigabit or 10-Gigabit Ethernet client network connectivity and bandwidth to the Isilon IQ storage cluster, enabling single-client high-performance HD Final Cut Pro workflows or multi-client lower-bandwidth workflows.

The table below (Figure 6) lists the stream count of various uncompressed HD formats per Accelerator-X 10-Gigabit Ethernet node and a 5-node 6000X storage cluster:

Video Format	Throughput per Stream	Ingest, Edit and/or Playout Streams per Accelerator-x
Uncompressed HD 4:2:2		
720p 60 8bit	105MB/s	5
1080i 50 10bit	132MB/s	4
720p 60 10bit	142MB/s	3
Uncompressed HD 4:4:4 RGB		
1080i 50 10bit	198MB/s	2
720p 60 10bit	211MB/s	2
1080i 60 10bit	237MB/s	1

Figure 6

This chart (Figure 7) displays CPU utilization per Accelerator-X node, per iQ6000X storage node, and an entire Isilon cluster for the tested streams presented above.

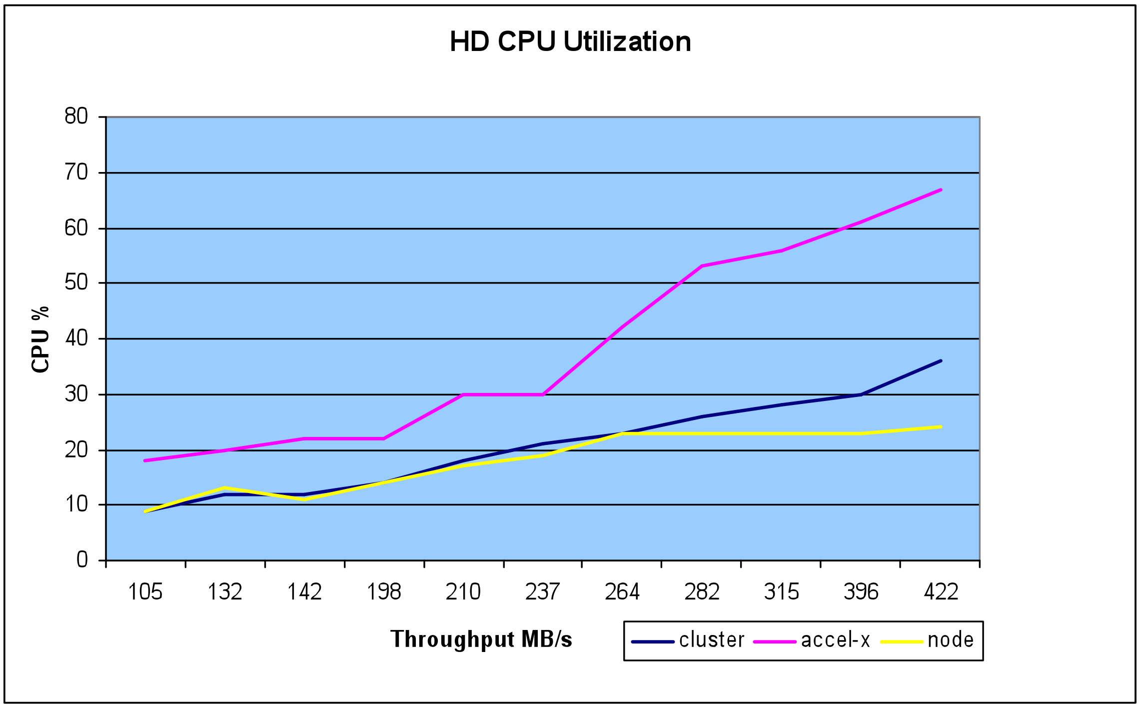


Figure 7

Sizing Isilon IQ Storage Cluster with Accelerator-X Nodes for Uncompressed HD

The chart (Figure 7) shows the following for 10-Gigabit Ethernet workflows

1. The accelerator-x CPU reaches 70% utilization at maximum load. This is normal since all client traffic is handled by the Accelerator-X node when serving uncompressed HD streams to / from the Isilon cluster at 130MB/s to 240MB/s throughput per stream.
2. The CPU load rises almost linearly as the aggregate throughput increases.
3. Performance caps at around 540MB/s throughput during which the accelerator CPU load reaches just below 70% utilization (2 uncompressed 1080i 10bit HD streams).
4. The CPU usage on each of the storage nodes rises moderately as they provide the disk I/O work behind the accelerator. There's plenty more storage node CPU available to handle additional accelerator-x I/O requests.
5. Cluster CPU rises near linearly as aggregate throughput increases.

The linear scalability behavior in bandwidth and CPU usage shows that by adding more accelerator-x nodes for bandwidth, and more storage nodes for disk I/O, more streams can be delivered by a single Isilon IQ cluster. These metrics provide a simple way to project the number accelerator-x nodes required to support a specific stream count per uncompressed HD video format and workflow. To size the number of accelerator-x nodes and storage nodes use the table above and do the following:

1. Divide the number of necessary uncompressed HD streams by the number of supported streams per accelerator-x node.
2. Since enough storage nodes are necessary to provide disk I/O behind the accelerator node at least 5 storage nodes are required for the first accelerator-x in the cluster and every subsequent accelerator-x node requires a base increment of 3 storage nodes in the cluster (so 6 nodes can support 2 accelerator-x nodes).

__NOTE__: The table and chart above are based on accelerator-x node with only 1 quad-core CPU and 4GB RAM and x-series storage nodes with only one dual-core Intel Xeon CPU and 4GB RAM. Newer accelerator-x models have two quad-core Intel Xeon CPU and up to 32GB RAM while newer x-series storage nodes have 1 quad core CPU. This should result in lower total and per node CPU utilization which will support additional streams per node.

Using the Accelerator-X with 1-Gigabit Ethernet

While the focus of this document is the overview of Isilon 10-Gigabit Ethernet support for uncompressed HD workflows, standard 1-Gigabit Ethernet Accelerator-X nodes may play a critical role in DV/SD workloads in the following scenarios:

1. Certain workloads such as rendering and compositing attempt to utilize maximum network bandwidth and consume extra storage CPU resources for handling the client network activity. This may affect individual storage nodes connected to clients from handling peer storage nodes I/O requests in a timely manner. As a result overall cluster performance may degrade. In such scenarios deferring client connections to an accelerator node may shield the storage nodes from direct client interaction and remove any localized unbalance load.
2. In certain conditions when a storage node drive failure occurs the node may become inaccessible for a few seconds beyond what normal video editing workloads can tolerate even when the data being edited does not reside on that node. By deferring client connection to an accelerator, clients do not interact directly with storage nodes and drive stalls on the storage node do not impact the client unless the data being requested resides on a drive within the storage node that is stalling.
3. Accelerator nodes employ a different caching model than storage nodes do. Accelerators cache entire files and maintain those in memory based on most recently used criteria. In workloads where the same file is being edited repeatedly this caching algorithm may result in higher response time and accelerate streaming of the file directly from the accelerator.
4. When a smaller amount of higher-density storage nodes are considered for replacing lower-density ones accelerator nodes can be added to compensate for the performance loss. Consider adding 2 accelerator nodes for every storage node missing from the new high-density storage configuration.
5. If higher density nodes are desired, in order to reduce the total number of storage nodes necessary to support the aggregate bandwidth, standard gigabit accelerator nodes can be added at a 2:1 ratio (accelerators to storage node) to compensate for the reduced number of storage nodes originally calculated. For example if a 10 x 6000x nodes was originally calculated to support a certain stream count and capacity requirements, a higher-capacity configuration of 6 x 12000x cluster may be used along with 8 x standard gigabit accelerator nodes to replace the 4 storage nodes missing from the higher-capacity configuration.

Mac Client Configuration Guidelines

Supported File Networks Protocols

Apple's Mac OS X 10.4 (Tiger), 10.5 (Leopard) and 10.6 (Snow Leopard) support mounting the Isilon volume via NFS and SMB.

With MacOS X 10.5 Leopard, Apple re-engineered the NFS protocol stack to extract better performance. Bandwidth testing with OS X Tiger 10.4.11 and NFS using a 10GbE interface card was capped at 150MB/s, while baseline testing with OS X Leopard 10.5.8 revealed the bandwidth on the 10GbE interface jumped to 540MB/s, a 3x performance increase. Performance continued to improve in OS X Snow Leopard 10.6, but not to the degree experienced from 10.4 to 10.5.

The Mac also supports the SMB file sharing protocol, which provides ~50% of NFS performance (55MB/s over 1-Gigabit Ethernet and 120MB/s over 10-Gigabit Ethernet). Therefore, SMB can be used for DV/SD workflows on Mac, whereas NFS is the protocol of choice of HD or other large I/O workflows.

> ***NOTE:*** *Isilon highly recommends upgrading Mac OS X hosts to the latest OS X version. Testing has been performed with OSX Leopard 10.5.3 through 10.5.8 and Snow Leopard 10.6.2 through 10.6.5.*

Network Interface 9000 MTU Setting
Increasing network MTU to 9000 (Jumbo Frames) reduces the overhead associated with IP packet fragmentation. This requires network cards and switches that support 9000 MTU setting. The built-in 1000base-T Ethernet ports on Intel MacPro, MacBook Pro and certain iMac models based on the Intel Core 2 Duo CPU support this setting.

> ***NOTE****: the Isilon OneFS 6.0 network profile will need to be set to 9000 MTU to support clients that are set to use the 9000 MTU setting.*

To change the Mac's MTU Network value to 9000, open *System Preferences* via the Apple menu, and choose the *Network* preference pane button. Select the green-lit Ethernet interface in the left column, and click the *Advanced* button in the lower right. In the Network Settings Pane that descends from the top of the System Preferences window, click on the far right *Ethernet* button and click on the *Configure* pop-up menu to change *Automatically* to *Manually*. The MTU pop-up menu will become available. Click on it to change the *Standard 1500* value to *Custom*, and type 9000 in the field that appears below it (in Snow Leopard, choose *Jumbo 9000*). Click the *OK* button to confirm this change.

> ***NOTE****: Isilon highly recommend using 9000 MTU with 10-gigabit network interface cards, but see below for special settings associated with using this MTU with 10Gbit PCIe cards in a MacPro.*

Mac PCI-Express Configuration

When installing a 10-gigabit network interface card on a MacPro 2006 (first-generation) client with PCI-Express bus, the PCI-Express bus lane distribution may need to be adjusted for optimal card performance. See *Appendix B* for the settings needed for this change.

Mac OSX NFS Mount Configuration – nfs.conf

When the Mac Finder is used to mount Isilon as a NFS volume, it can introduce significant overhead and limitations that impact NFS performance. However, as of Mac OS X 10.5 and 10.6, Isilon NFS shares can be mounted using the Mac Finder (via the *Go* menu -> *Connect to Server*) after NFS mount parameters are modified by creating a *nfs.conf* file.

Mac OS X 10.5 and 10.6 clients do not include a nfs.conf file within the /etc directory. By creating a nfs.conf file within the /etc directory of Mac OS X, the Finder's mount parameters are tuned for performance for Final Cut Pro Mac clients using Isilon via NFS.

To create nfs.conf settings that are applied via the *Connect to Server* mount method, the nfs.conf file should contain this line:

```
nfs.client.mount.options=nfssvers=3,tcp,async,nolock,rw,rdirplus,rsize=65536,wsize=65536
```

For additional performance, latency in NFS write operations can be reduced further. This is accomplished by modifying the nfs.conf file to include this additional line:

```
nfs.client.allow_async=1
```

Async requires the administrator to go into the Isilon WebUI and modify NFS File Sharing parameters. See **Isilon IQ Cluster Configuration Guidelines** for this information.

> ***NOTE:*** *The nfs.client.allow_async=1 line with a slight risk of the chance of losing a write, if a node were to go down. Contact Isilon Support for further clarification of the impact of using the async option.*

Mounting the Isilon NFS share

In the Mac Finder, click on the Go menu and select Connect to Server. In the dialog window that appears, type ***nfs://<node>/ifs/data***.

> ***NOTE:*** *<node> can be either the Node DNS name or IP address, or the Isilon SmartConnect address equivalent*

To mount the Isilon NFS share automatically on login, follow these steps:

1. Modify the Finder Preferences, to show Connected Servers on Desktop.
2. Open System Preferences -> Accounts preference pane. Click on Login Items, and drag the icon of the Isilon NFS mount into the Login Items window.

TCP Memory Buffers

It is possible that TCP memory buffer settings on the Mac clients are adding network latency. Increasing these settings will help as long as there are no lost packets, otherwise significant latency problems may arise. A reasonable value for TCP buffer size setting is somewhere around the amount of port buffering on the intermediate switches and cluster. This setting can go up to 1MB on both the cluster and the Mac client, if the intermediate switches support it, or 32k or so for lower end switches (for example, the high-end Cisco 3750G switch has a TCP buffer value of 512k per port). On the Isilon cluster the default setting is 128k for both send and receive buffers. It may help to change those setting on the Mac client to the same value (measured in bytes):

```
# sysctl -w net.inet.tcp.recvspace=131072
# sysctl -w net.inet.tcp.sendspace=131072
```

On Mac clients with a 10-Gigabit Ethernet network interface card, use the following sysctl settings:

```
# sysctl -w kern.ipc.maxsockbuf=4194304
# sysctl -w net.inet.tcp.mssdflt=8960
# sysctl -w net.inet.tcp.sendspace=524288
# sysctl -w net.inet.tcp.recvspace=524288
```

To make this setting permanent, add everything after the "-w" to

/etc/sysctl-macosxserver.conf (Mac OS X Server) or /etc/sysctl.conf (Mac OS X Client).

Mac OSX Client NFSIOD Threads

(Deprecated in Mac OS X 10.6 because the nfsiod value defaults to 16. These settings only apply for Mac OS X 10.5 or older)

A factor that can affect MacOS NFS performance is the number of nfsiod threads. These are user space threads that provide the interface between the NFS kernel module and the user space application sending and receiving data over NFS. Adding more threads means that more NFS transactions can execute concurrently (including read-ahead and write-behind operations).

To increase this value on Mac OS X, edit the file /System / Library / StartupItems / NFS / NFS, and change the line that reads:

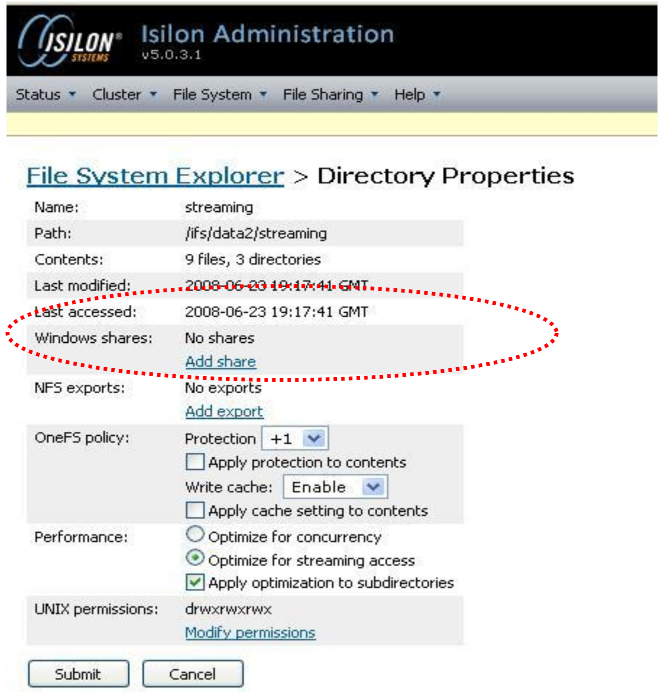
nfsiod -n 4.

Replace the 4 with 8 or more, followed by manually killing and re-running nfsiod by running:

‘killall nfsiod; nfsiod –n 8’

To increase this value on Mac OS X, run this command:

‘nfsiod –n 16’



Streaming Workload Optimization

In OneFS 5.0 a new file layout was introduced to enhance performance for streaming workflows. Once applied, more disks are used in parallel to write a file. This layout can be applied to any directory so all files within that directory benefit from this streaming optimized mode.

From the Isilon WebUI, use the File Explorer to navigate to the desired directory and select the “optimize for streaming” radio button from the Performance section. To apply this setting to all sub-directories click the check box indicated to the right.

Read Optimization for Multi-File Workflows

Certain media workflows such as 2K scanning use video formats such as DPX in which each file represents one video frame. These 10-bit files are typically 12MB in size and numbered in sequence based on their order in the video timeline. By default, OneFS employs a prefetch mechanism to read ahead data within an open file but it does not prefetch data beyond the end of the open file. To optimize read performance when reading a video stream comprise of a sequence of DPX or similar files, OneFS can be directed to prefetch data from a sequence of files based on a the appearance of on the numbering scheme embedded within the file names. To turn on this feature for all sub-directories in streaming mode use the following syctl:

```
sysctl efs.bam.fnprefetch.enable_streaming=1
```

To turn on this feature for all sub-directories in concurrent mode use the following syctl:

```
sysctl efs.bam.fnprefetch.enable_concurrent=1
```

These sysctls should only be applied in accelerator-x nodes on a node by node basis. To make these sysctls permanent locally on a specific node, edit the file /etc/sysctl.conf and add either or both of the following lines:

```
efs.bam.fnprefetch.enable_streaming=1
efs.bam.fnprefetch.enable_concurrent=1
```

NOTE: *using streaming mode or multi-file prefetch mode does not provide any extra benefit on client connected to storage nodes or standard accelerator nodes over 1-Gigabit Ethernet connections. These setting should only be applied to Accelerator-X nodes in multi-file high-performance streaming read workflows.*

Async Option in nfs.conf
The administrator must go into the Isilon WebUI on the cluster, then to File Sharing -> NFS and set the Write Commit Behavior to *Asynchronous*.

> ***NOTE:*** *The cluster does not support asynchronous writes from clients when SmartConnect with dynamic IP addresses is used. Instead, the client should mount a node directly and not through SmartConnect.*

TCP Inflight Delay
Turning off the following sysctl on the cluster or node that is handling client requests may help read speeds. Run the following command on a node to disable this sysctl:

    # sysctl net.inet.tcp.inflight.enable = 0;

Run the following command to disable this sysctl on all nodes:

    # isi_for_array -s "sysctl net.inet.tcp.inflight.enable=0"

To make this setting permanent edit the file /etc/mcp/override/sysctl.conf on any node on the cluster and add the following line:

    net.inet.tcp.inflight.enable=0

TCP delayed ACKs
Turning off the following sysctl on the cluster may help read speeds for NFS operations with large read and write buffer size rwsize=128K or higher). Run the following command on a node to disable this sysctl:

    # sysctl net.inet.tcp.delayed_ack=0

Run the following command to disable this sysctl on all nodes:

    # isi_for_array -s " sysctl net.inet.tcp.delayed_ack=0"

To make this setting permanent edit the file /etc/mcp/override/sysctl.conf on any node on the cluster and add the following line:

    net.inet.tcp.delayed_ack=0

Cluster Restriper Scheduling
In high load video processing, restriper jobs may cause contention related latency leading to dropped frames. Currently non-critical restriper jobs run on a schedule, hard-coded into the restriper. It would be helpful to configure the restriper service to run the following maintenance jobs only on off-hours schedule: autobalance, mediascan, collect. This can be done by setting up cron table entries to enable/disable the restriper service on a scheduled basis. The cron table entries and enable/disable script it runs are available by request from support.

> ***NOTE:*** *The restriper flexprotect job will run regardless of any cron table entries in case of drive or node failure.*

Supporting Large Files on Mac Clients over NFS (for legacy OneFS 5.0 or earlier)

As an NFS client Mac OSX 10.5 and 10.6 exchanges protocol information with the NFS server to discover the file system capabilities such as maximum file size and file system size. Mac OSX relies on a specific response from Mac OSX NFS servers to allow applications to create large files. As a result finder copy operations may fail when coping files over 4GB in size, or Final Cut Pro will break a capture stream to multiple files each 2GB in size.

> *NOTE: maxfsize does not need to be adjusted in OneFS 5.5 or 6.0 to support large file sizes.*

To modify this behavior and allow for any arbitrary file size copy and capture over NFS, Isilon IQ NFS server should be configured to emulate the behavior of Mac OS NFS servers.

This setting can be applied permanently to the entire cluster by adding the following line to the file /etc/mcp/override/sysctl.conf:

```
vfs.nfsrv.macos_maxfsize=1
```

Cluster SAMBA Performance Tuning
The following recommendations apply specifically to SMB file sharing.

SMB signing
On Isilon IQ clusters, SAMBA signing is turned on by default. A digital signature is placed into each SMB message and verified by both client and server. SMB signing is use to protect against “man in the middle” security attacks, but some Mac clients do not support this type of encrypted connection. To remove SMB signing on the cluster, edit the file /etc/mcp/override/smbd.xml file on any node on the cluster, and add the following XML tag immediately after any line that ends with </add-tag>:

```
<add-tag id="smbdglobal">
        <field name="client signing" value="disabled"></field>
</add-tag>
```

Once you have made this change, wait 60 seconds for it to propagate to all other nodes.

> ***NOTE:*** *Cluster Samba service will restart after this change is made.*

SMB Alternate Data Streams
Alternate Data Streams (ADS) enables the Isilon to store data related to a file without appearing as part of that file to other systems. This setting may include file metadata, "resource forks" for Macintosh clients, or other information. In certain circumstances, ADS may cause enumeration delays for applications – in which case hiding them would improve performance. However, it is not advised to disable support for ADS, for it doubles the number of Mac-specific file metadata objects in the directory. Nonetheless, to disable support for ADS for all shares in a cluster, edit the file /etc/mcp/override/smbd.xml on any node on the cluster, and add the following XML tag to the file immediately after any line that ends with </add-tag>:

```
<add-tag id="smbdglobal">
        <field name="ignore named streams" value="yes"></field>
</add-tag>
```

Once you have made this change, wait 60 seconds for changes to propagate to all nodes.

> ***NOTE:*** *Cluster Samba service will restart after this change is made.*

Conclusion

The new Isilon IQ cluster was designed from the ground up to address the challenges of today's multiple Final Cut Pro application and workflow scale-out storage needs. Coupling the Isilon IQ Accelerator-X platform with the latest featureset of OneFS 6.0 and the network advancements in Mac OS X 10.6, a wide range of application workflows from DV to uncompressed HD can now be captured, edited, transcoded, processed and played out directly from the Isilon IQ cluster – leveraging the benefits of a single filesystem.

In addition, the FCP shared-editing capabilities on Isilon iQ provide numerous IT benefits not possible on other FCP-specific shared edit platforms such as Terrablock, EditShare, Unity and Xsan. These benefits include the following IT-specific features:

1. SmartPools for data tiering across different cost-optimized disk storage layers within a single Filesystem;
2. InsightIQ for storage analytics;
3. Snapshot IQ for data restoration;
4. FlexProtect for N+ data protection that goes beyond RAID 6 protection, at the directory or file level;
5. No cost for client connectivity; and
6. SmartConnect Advanced, for data-rate shaping using the NFS protocol.

About Isilon Systems
Isilon Systems (a division of EMC) is the worldwide leader in clustered scale-out storage systems and software for digital content and unstructured data, enabling enterprises to transform data into information - and information into breakthroughs. Isilon's award-winning family of IQ clustered storage systems combines Isilon's OneFS® operating system software with the latest advances in industry-standard hardware to deliver modular, pay-as-you-grow, enterprise-class storage systems. Isilon's scale-out clustered storage solutions speed access to critical business information while dramatically reducing the cost and complexity of storing it. Information about Isilon can be found at **http://www.isilon.com**.

Appendix A: Lab Test Configuration

The performance guidelines in this document are based on testing conducted using the following hardware gear and software suite.

Isilon IQ Cluster	FCP Clients	Playout Server	Network
(5) iQ 6000x 1 quad-core Xeon 4 GB ram OneFS 6.0.1 **(1) Accelerator-X** 2 quad-core Intel Xeon 4 GB RAM Chelsio 10GbE OneFS 6.0.1	**(2) Mac Pro** 2 quad-core Intel Xeon 8GB RAM Built-in 1GbE NIC 10GbE Myricom NIC 1 AJA Kona LHi card Mac OSX 10.6.3 FCP Pro 7.0.3 QuickTime 10.0	**(1) Dell Precision 490** 2 dual-core Xeon 3 GB RAM 1 GbE NIC 1 Myricom 10GbE 1 Decklink Pro HD I/O Windows XP 32-bit Hummingbird Maestro NFS client Decklink MediaExpress	**Cisco 3750** 1 GbE switch 9000 MTU **Fujitsu XG700-CX4 Copper** (10) 10GbE ports. 9000MTU

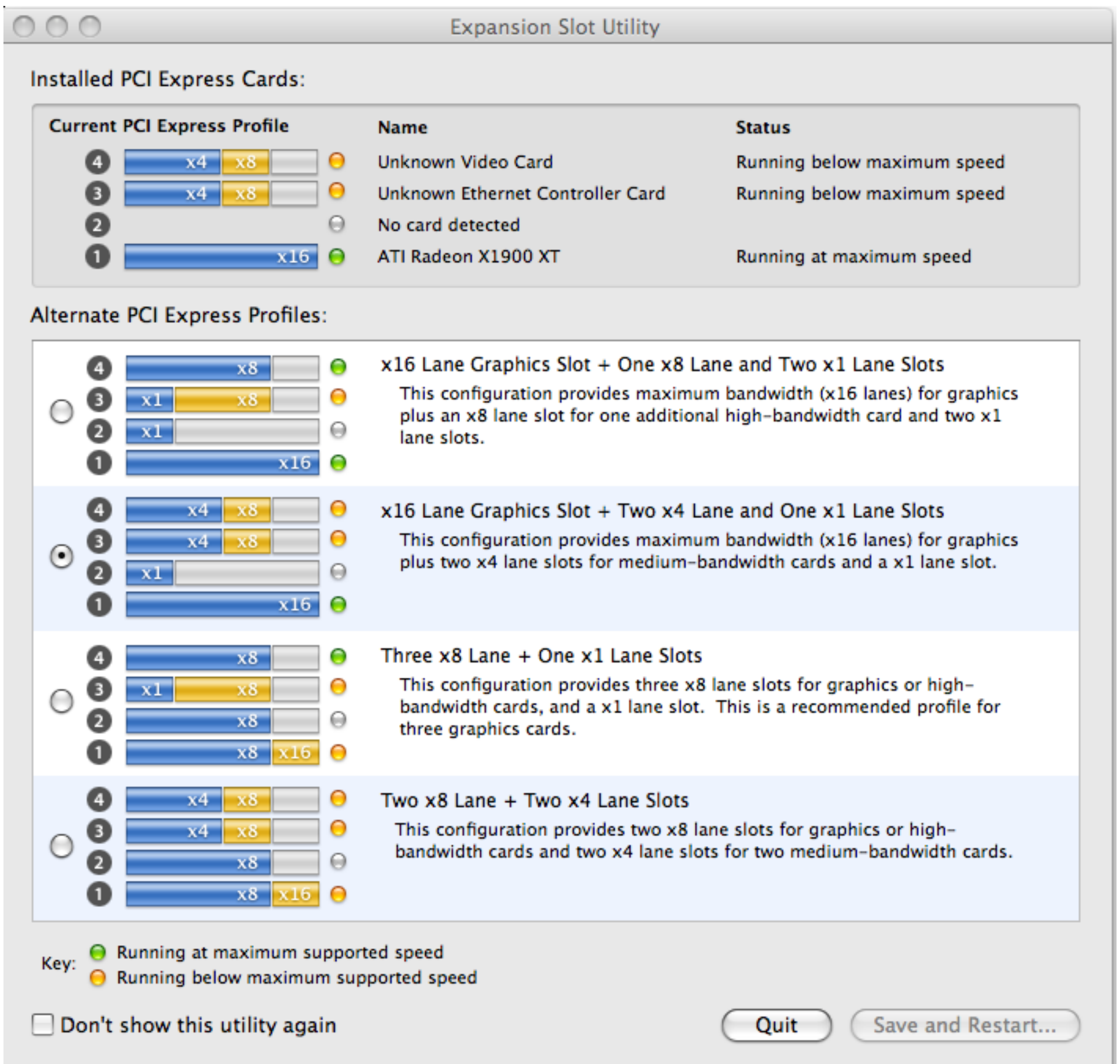
Appendix B: PCIe 10GbE Bus Speed Settings for MacPro 2006

Mac PCI-Express Configuration
When installing a 10-Gigabit Ethernet network interface card on a MacPro 2006 (first-generation) with PCI-Express bus, the bus lane distribution may need to be adjusted for optimal performance.

To do so run the following Mac utility:

/System/Library/CoreServices/Expansion\ Slot\ Utility.app/Contents/MacOS/Expansion\Slot\ Utility –autolaunch

Once the following screen appears, select the second option from the top to ensure that the 10-Gigabit Ethernet network card and optional video I/O card are allocated the proper amount of PCI lanes. Both cards should be installed in the top two slots of the MacPro 2006 system.



***Note:** It is unnecessary to run this utility on any MacPro 2007, 2008, 2009 or 2010 system.*