Value Added
Reseller
Professional Video

Preparing for Final Cut Pro X

A primer on the Apple's latest video technology
and how to optimize your system for Apple FCP X

Torrey Loomis
Silverado Systems, Inc.

May 2011

Contents

Introduction & Background

For years, people asked questions about how to build Final Cut Pro systems:

- *What kind of Mac should I edit with?*
- *Do I need a tower?*
- *How much RAM should I buy?*
- *What kind of storage space do I need?*
- *What graphics card should I get?*

This has been a perplexing issue since Final Cut Pro v1.0 was released. Final Cut Pro has always been a 32-bit program. That means by itself, FCP does not use more than 4 GB of RAM. However, that doesn't mean that the entire system couldn't benefit from more RAM.

That begs the bigger question--how much power (RAM, graphics cards, CPU cores, etc...) can you apply to an FCP system before you're wasting money? Is there a stopping point? Its never been totally obvious where the cutoff is located at which FCP simply can't perform better. However, there is a certain point where the current version of FCP just can't render faster, display larger resolutions, or stack more tracks without dropping frames. No amount of more RAM or CPU power can push you past that point in Final Cut Pro 7.

However, Final Cut Pro X changes all of that--with a little help from Mac OS X 10.6 Snow Leopard.

Let me be quite clear: the sole purpose of this paper is to convince you to purchase the most powerful Mac system you can possibly buy.

The reason? For the first time, Final Cut Pro will directly take advantage of all your cores, all your RAM, and employ your graphics cards for computational performance never seen before in a non-linear editor. There are some major changes under the hood of the new software that pairs significantly faster performance with respectively more powerful hardware.

This document will cover topics such as Grand Central Dispatch, 64-bit code, OpenCL, and other implementations that directly empower the new Final Cut Pro X codebase. These technologies will have a direct impact on your editing experience.

Each page will discuss a particular topic and then end with a summary and a recommendation. The summary will suggest "what does this feature do to enhance my editing experience better" while the recommendation will tell you what you can do to maximize that particular feature in your own system.

There are no performance numbers available to test these metrics: Final Cut Pro X is not released at the time of writing. This document will be updated with system performance analysis once Final Cut Pro X is released.

64-bit Computing in Mac OS X

It seems like it was ages ago, but when it was introduced in 1999, Final Cut Pro initially ran on Mac OS 8.5 with seemingly antiquated system requirements:

- Power Macintosh G3 266 Mhz (300 Mhz required for DV)
- Mac OS 8.5 or later
- 128 MB of RAM
- CD-ROM drive
- 6.0 GB AV rated drive
- Color display
- ATI built-in video support on G3 models (required for DV)
- QuickTime-compatible card for capturing video
- FireWire or device control cable

Also--the initial release of Final Cut Pro was not supported on iMacs or PowerBook systems.

Fast forward a decade to Final Cut Pro 7 and Mac OS X 10.5 Leopard. Minimum system requirements increased to the following:

- Mac computer with an Intel processor
- 1.0 GB of RAM (2.0 GB of RAM recommended when working with compressed HD and uncompressed SD sources; 4.0 GB of RAM recommended when working with uncompressed HD sources)
- ATI or NVIDIA graphics processor.
- 128 MB of VRAM
- Display with 1280-by-800 resolution
- Mac OS X v10.5.6
- QuickTime 7.6
- DVD drive
- 4.0 GB of disk space

While the system requirements significantly increased between FCP version 1.0 and 7.0--Final Cut Pro was still limited to using 4.0 GB of RAM because of the underpinnings of some of its original 32-bit Carbon code and the limitations of all Mac operating systems up through--and including--Mac OS X 10.5.

Mac operating systems have come a very long way in the last 20 years. 32-bit memory addressing was released in Mac OS 6, while Mac OS 7 introduced a true 32-bit system, support for virtual memory, and QuickTime. Mac OS 8 introduced a feature called “multithreading” which allowed the system to work with files simultaneously. The HFS file system was introduced in Mac OS 8.1 while Sherlock (an early form of Internet search) was rolled out in Mac OS 8.5.

Mac OS 9 introduced features such as Software Update and support for multiple user accounts. Mac OS 9.22 was the last version of the operating system released before Apple’s transition to OS X. The max memory that OS 9 could handle was 1.5 GB and each application could only address max of 1.0 GB RAM.

With the introduction of Mac OS X, Apple started peeling away the restrictions of the prior OS. While Mac OS X 10.0 Cheetah was essentially the initial release of OS X, many people waited until Mac OS X 10.1 Puma was rolled out before even trying OS X.

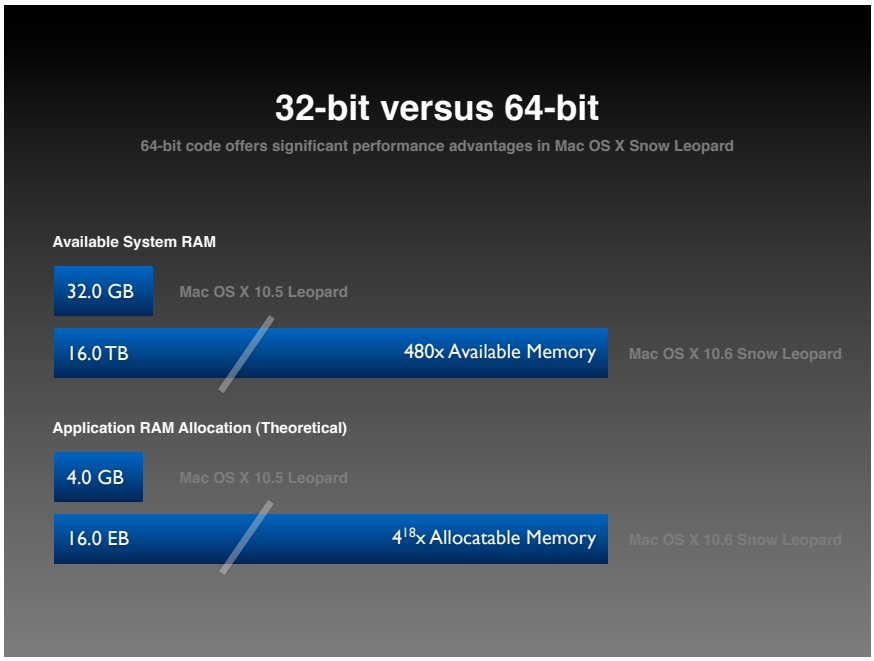
Releases from Mac OS X 10.2 Jaguar to Mac OS X 10.4 Tiger represented the primary releases that were broadly accepted as usable platforms by the Apple community. There were a tremendous amount of features rolled out during these releases:

- Bonjour
- CUPS (Common Unix Printing System)
- Quartz Extreme
- Journaled file system
- Fast User Switching
- Exposé
- Font Book
- Spotlight
- Dashboard

However, the majority of OS X at this point was still 32-bit dependent. Mac OS X 10.4 started to transition over to 64-bit processes and servers with a 64-bit layer of Unix, but all apps, kernel extensions and drivers were still 32-bit. You could spawn a 64-bit process from within a 32-bit program, but it was not possible to run a 64-bit native GUI. Under Mac OS X 10.5, a layer of 64-bit Cocoa was introduced that allowed native 64-bit apps to be run--not just spawned and controlled by a 32-bit app.

While many people believed that Snow Leopard was mostly some form of “maintenance release” of Mac OS X 10.5 Leopard, it was actually a massive upgrade to potential system capabilities of each Mac. Under Snow Leopard, the system itself was upgraded to full 64-bit. Further, all kernel extensions and drivers were rewritten as native 64-bit--even the kernel itself is 64-bit.

The change was monstrous.



The above chart shows just how massive a performance leap arrived with Mac OS X 10.6 Snow Leopard. Whereas the max system RAM available under Mac OS X 10.5 Leopard was 32.0 GB, it is now possible to address 16.0 TB of physical memory under Mac OS X 10.6 Snow Leopard. That is 480x more memory than Mac OS X 10.5 Leopard (and presently impossible with current RAM capacities).

And as backwards as this may sound, applications under Mac OS X 10.6 Snow Leopard can theoretically address up to 16.0 EB (exabytes) of memory. This is significantly more memory than could ever physically be installed inside even the largest Mac Pro.

In practice--in a pure 64-bit system--there is no possible way to create a maxed-out Mac system that a properly coded application that uses 64-bit memory addressing cannot fully exploit.

Another way that your system will use massive amounts of RAM efficiently is planted in Final Cut Pro X's new hooks into OpenCL processing. While it will be further discussed in a later section, its important to note that Mac OS X 10.6 Snow Leopard allows applications to make extensive use of graphics processing unit (GPU) resources using OpenCL. OpenCL will shuttle large amounts of data between your GPU and your main memory, so having more memory will optimize your system's ability to effectively utilize as much OpenCL power as possible.

The OS resources are there for well-written and well-coded programs to grab every last bit of memory made available. In a pure 64-bit environment (system, drivers, and apps) your memory limitations are no longer the operating system or the application, but the amount of physical RAM you can throw in a box.

The performance restraints of past 32-bit system architecture have been thrown out under Mac OS X 10.6 Snow Leopard. Performance will continue to improve under Mac OS X 10.7 Lion, but the capability to write extremely efficient 64-bit code exists now under Snow Leopard.

The secret to unlocking this power is in the software. It takes extraordinary code to take full advantage of the massive amounts of RAM that are shipping with today's Mac systems. However, Final Cut Pro has been stuck in 32-bit code for years. With the development of clean 64-bit native code in Final Cut Pro X--Apple has provided a means for editors to access significantly more than the 4.0 GB of RAM that Final Cut Pro 7 is currently uses.

Summary

Because of the change that 64-bit architecture has brought to the Mac under Mac OS X 10.6 Snow Leopard, apps that are properly coded and optimized for 64-bit can realize tremendous gains in memory allocation and utilization. While typical Mac users saw only minor changes upgrading from Mac OS X 10.5 Leopard to Mac OS X 10.6 Snow Leopard, the changes "under the hood" were huge. With the advent of optimized modern applications like Final Cut Pro X, professional users will have access to massive amounts of computing power that were previously untapped in their systems.

Recommendation

Install as much RAM as you can afford–Final Cut Pro X will use it. Having 64 GB of RAM isn't just a dreamers ideal--it will soon become a fairly common setup. On top of that, OpenCL means that your CPU gets much more direct assistance from your GPU, so adding more RAM and faster GPU's will also speed up the professional desktop as more of the computer contributes to the overall tasks at hand.

Grand Central Dispatch

In the past, hardware companies threw resources at making processors faster in terms of megahertz (MHz) and then gigahertz (GHz). However, there was a limit to how fast this speed race could continue its acceleration. An excellent example of this is posted on the Power Mac G5 Wiki page:

"...Steve Jobs stated during his keynote presentation that the Power Mac G5 would reach 3 GHz "within 12 months." This would never come to pass; after three years, the G5 only reached 2.7 GHz (or dual-core at 2.5 GHz) before being replaced by the Intel Xeon-based Mac Pro, which included processors with speeds of up to 3 GHz, and after three years is presently at 3.33 GHz."

Developers realized that while super-fast single-core processors had limited growth capability, one area of growth that had not been fully exploited were large collections of processing cores on a single chip. Also known as multiple cores or multicore, these collections of processors started to show up in Mac systems with the advent of the Power Mac G5 and became even more commonplace on Mac systems after the transition to Intel processors.

Presently, there are no shipping Mac systems that don't include some form of multicore processors. Entry level MacBook, MacBook Air, and Mac mini systems have Intel Core 2 Duo chips. Apple iMacs feature Intel Core i3 and Intel Core i5 technology--and the most powerful iMac features a quad-core Intel Core i7 chip.

In Apple's professional line-up, MacBook Pro systems feature dual Intel Core i5 and Intel Core i7 chips along with a quad-core Intel Core i7 offering. And Mac Pro systems start at 2.8GHz quad-core Intel Xeon and 3.33 Ghz 6-core Intel Xeon in a single processor "Nehalem" configuration and scale up to 2.93 Ghz 12-core Intel Xeon in a dual "Nehalem" configuration.

Before jumping into Grand Central Dispatch, lets look at two major distinctions between Apple's consumer and professional lines. There are two performance enhancing technologies from Intel that are found in both Apple's MacBook Pro and Mac Pro systems: Turbo Boost and Hyper-Threading.

Intel Turbo Boost is described on the Intel Turbo Boost Wiki page:

"...Turbo Boost...enables the processor to run above its base operating frequency via dynamic control of the CPU's "clock rate". It is activated when the operating system requests the highest performance state of the processor...an open standard supported by all major operating systems; no additional software or drivers are required to support the technology. The design concept behind Turbo Boost is commonly referred to as "dynamic overclocking"

"...When workload on the processor calls for faster performance, and the processor is below its limits, the processor's clock will increase the operating frequency in regular increments as required to meet demand. Frequency increases occur in increments of 133 MHz for Nehalem micro-architecture processors and 100 MHz for Sandy Bridge micro-architecture processors."

The Wiki page also brings up an extremely relevant concern about Turbo Boost:

"While Turbo Boost has the potential to speed up single threaded tasks that are unable to otherwise take advantage of the additional cores, it is very rare to see the full advantage in practice. At issue is the need for two or three cores to be inactive to reach the two or one core active turbo speeds; [some operating systems such as] ...Windows will take a single thread and run 25% of it on each of four cores instead of putting it all on one core...since it is a single thread, there is only one core active at a time, the other cores need time to go to sleep and allow the running core to boost up. As a result, the single-core speed is not seen and the two-core speed is rarely seen..."

This is a paradox that applies to current Mac apps that aren't coded properly to take advantage of multiple processors:

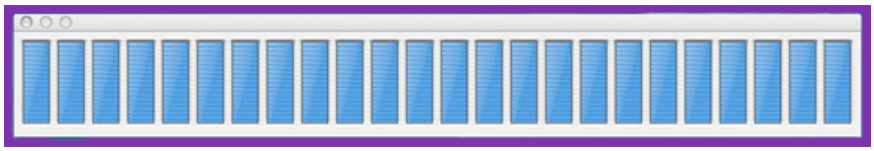
- If your app is coded to use multiple cores--but does so in an inefficient way--you'll never tap the extra speed made available under Turbo Boost because each core will not reach the threshold *required to trigger* a boost.
- If your app is not coded to use multiple processors--you may have a single core that triggers a boost, but you are missing out on the power available throughout the rest of the cores.

The key to solving this Turbo Boost paradox is having an underlying technology that can adaptively spread tasks efficiently across multiple cores as equally as possible. Apple's Grand Central Dispatch helps by load-balancing tasks across available cores. As core speed is taxed, the boosting functionality can kick in equally across cores making processing much more efficient.

The second technology that distinguishes Apple's professional line is Intel's Hyper-Threading:

"Hyper-threading is an Intel-proprietary technology used to improve parallelization of computations (doing multiple tasks at once) performed on PC microprocessors. For each processor core that is physically present, the operating system addresses two virtual processors, and shares the workload between them when possible. Hyper-threading requires not only that the operating system support multiple processors, but also that it be specifically optimized for HTT."

Essentially, Intel's Hyper-Threading allows each core to process two software threads at once. This presents to Mac OS X 10.6 Snow Leopard as double the amount of available cores. For example, a 12-core Nehalem-based Mac Pro (represented visually below in Activity Monitor) shows 24 available cores for processing.



While 24 cores fully pegged at 100% utilization is rare now, it will become extremely commonplace with apps like Final Cut Pro X that heavily utilize Grand Central Dispatch.

So what exactly is Grand Central Dispatch? Apple describes it this way:

"Grand Central Dispatch is a revolutionary, pervasive approach to multicore processing. GCD shifts the responsibility for managing threads and their execution from applications to the operating system. Mac OS X Snow Leopard provides APIs for GCD throughout the system and uses a highly scalable and efficient runtime mechanism to process work from applications. As a result, programmers can write less code to deal with concurrent operations in their applications, and the system can perform more efficiently."

Apple continues...

"GCD gives developers a simple way to describe the different pieces of work that your applications need to do and an easy way to describe how those pieces might be dependent upon one another. Units of work are described as blocks in your code, while queues are used to organize blocks based on how you believe they need to be executed. By using blocks and queues, you no longer need to worry about threads, thread managers, or locking data, making an application's code easier to understand. You can simply let the system manage the work queues and execute the blocks for optimal performance."

With the appropriate changes to code, developers can author apps that bundle workloads into blocks--which are extensions of C, C++, and Objective-C. These blocks can be organized for processing in queues. Queues are scheduled, executed, and managed by Grand Central Dispatch at the operating system level. That means developers no longer have to worry about optimizing individual apps to handle multicore work so they don't bump into performance boundaries of other applications. Using tools provided at the OS level, developers can create excellent multicore aware apps that are managed efficiently by the underlying operating system.

Summary

Grand Central Dispatch is a load-balancing technology that takes properly coded applications and effectively balances performance across multiple cores. Since performance is balanced, full utilization of all cores can occur. Intel's Hyper-Threading technology allows Mac OS X 10.6 Snow Leopard to assign twice as many tasks to available cores--effectively doubling the amount of processing power available in systems that support Hyper-Threading. And as the threshold of core speed is approached, Intel's Turbo Boost technology provides another level of speed by allowing cores to dynamically overclock themselves to a higher level of performance.

Recommendation

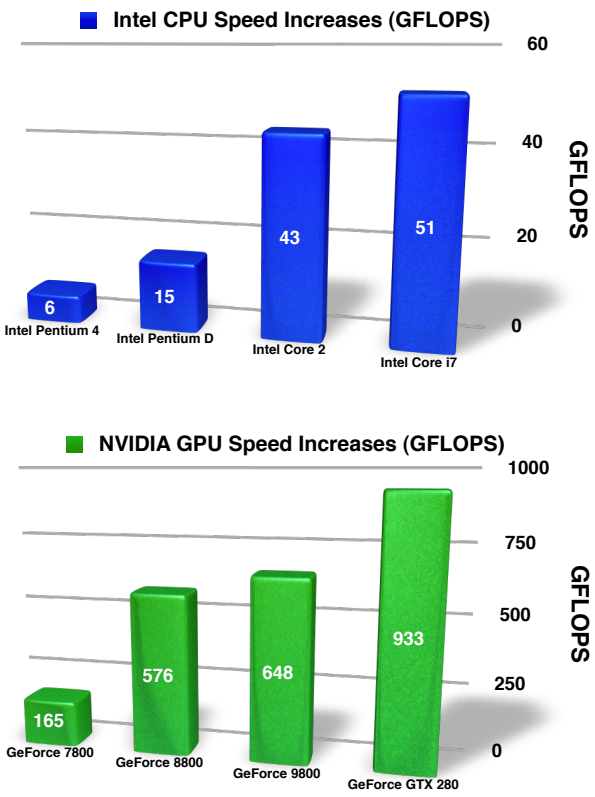
You can fully leverage the maximum amount of power available via Grand Central Dispatch by using Apple systems that support Intel Hyper-Threading and Turbo Boost (presently 2009 and later Mac Pro and 2011 MacBook Pro systems). To get the highest performance you'll need to make sure the apps you are running are Grand Central Dispatch optimized--such as the new Final Cut Pro X.

Finally, when making a choice on systems--getting a system with the maximum number of cores available for your budget will increase your performance by the greatest factor. If you are running a pre-2009 Mac Pro or pre-2011 MacBook Pro, consider upgrading to a new system to take advantage of the significantly higher amounts of power made available on those architectures.

OpenCL

What would you say if we suggested you had a supercomputer inside your Mac Pro?

No--we're not talking about your central processors. While CPU performance has been leveling off, GPU performance is still readily increasing. An example of the disparity of performance increases between CPU and GPU performance can be seen respectively in the progression of speed improvements made by Intel in their CPU's and NVIDIA in their GPU's.



The chart to the right shows increases in speed between Intel's Pentium 4 series and the Intel Core i7 series.

In contrast, NVIDIA GPU performance went from 165 GFLOPS in the GeForce 7800 series GPU's to 933 GFLOPS in the GTX 280 GPU series.

In the same general time frame, CPU performance was able to add 45 GFLOPS of performance whereas GPU performance increased 768 GFLOPS.

This is a trend that was noticed by Apple and they decided to come up with a new technology called OpenCL that allows general computing applications to take advantage of the advanced power that sat--sometimes largely unused--in the confines of graphics processing units. Before we look at OpenCL in detail, lets look at its software cousin OpenGL.

You might be familiar with OpenGL--otherwise known as Open Graphics Library. As described on the OpenGL Wiki page:

"...a standard specification defining a cross-language, cross-platform API for writing applications that produce 2D and 3D computer graphics. The interface consists of over 250 different function calls which can be used to draw complex three-dimensional scenes from simple primitives. OpenGL was developed by Silicon Graphics Inc. (SGI) in 1992 and is widely used in CAD,virtual reality, scientific visualization, information visualization, and flight simulation. It is also used in video games..."

OpenGL is a tool that allows developers to write sophisticated code that can be uniformly presented to different GPU platforms from multiple vendors. It prevents a significant amount of duplicate work because you can code once and be assured that OpenGL-compliant equipment will perform similarly regardless of manufacturer. OpenGL hardware is required to meet certain minimum standards (supporting the OpenGL feature set) in order to be listed as a compliant card--so developers are assured of a base standard of performance.

This is great for developers who code for graphics-specific applications. However, prior to OpenCL, the GPU in your system could not be repurposed for anything else. If you wanted to process compute intensive tasks that were not connected to something graphics related that could be coded for OpenGL, you were generally left with using the CPU's as your only processing option.

OpenCL was created to allow general purpose applications the ability to shift normally CPU-driven tasks off to the GPU. From the OpenCL Wiki page:

"...OpenCL (Open Computing Language) is a framework for writing programs that execute across heterogeneous platforms consisting of CPUs, GPUs, and other processors...It has been adopted into graphics card drivers by both AMD/ATI ...and NVIDIA. OpenCL gives any application access to the Graphics Processing Unit for non-graphical computing. Thus, OpenCL extends the power of the Graphics Processing Unit beyond graphics..."

The grand takeaway from this resultant shift in computing power is that properly coded apps that use OpenCL can move a massive amount of compute-intensive tasks off the CPU and onto the GPU. An example of this was the recent announcement by Blackmagic Design that DaVinci Resolve 8 would incorporate OpenCL code into the new release. The result of this was the certification of heretofore previously unsuitable systems as "certifiably qualified" systems on which to run DaVinci Resolve:

"Advanced OpenCL image processing has been incorporated into DaVinci Resolve 8...this allows a broader range of GPUs to be used for real time processing up to 1080 HD resolutions. OpenCL based processing, while not as powerful as CUDA processing also used on DaVinci Resolve, does allow a much wider range of computers that can be used for color grading..."

Summary

Apple summarizes OpenCL well in their OpenCL Technology Brief:

"...the trends in GPU designs offered an incredible opportunity to take the GPU beyond graphics. All that was needed was a non-graphics API that could engage the emerging programmable aspects of the GPU and access its immense power. OpenCL is that technology, delivering the means for any application to access the supercomputer- like performance of the modern GPU."

Recommendation

You need three things to make OpenCL work for you:

- Operating system with OpenCL support
- Supported graphics card
- Applications that are optimized for OpenCL

OpenCL 1.0 was rolled into the release of Mac OS X 10.6 Snow Leopard, so if you are planning on running Final Cut Pro X with OpenCL, you must upgrade from Mac OS X 10.5 Leopard to Mac OS X 10.6 Snow Leopard. (Its possible that Final Cut Pro X may not even run on Mac OS X 10.5 Leopard due to the tight integration between FCP X and other underlying features of Mac OS X 10.6 Snow Leopard such as Grand Central Dispatch and large 64-bit memory allocations.)

Check out the following page for graphics cards recommendations.

Graphics Card Suggestions

In order to get the maximum benefit from OpenCL optimizations built into Final Cut Pro X, you'll want to utilize a GPU that is qualified for OpenCL.

NVIDIA Qualified GPU Cards

- GeForce 320M
- GeForce GT 330M
- GeForce 9400M
- GeForce 9600M GT
- GeForce 8600M GT
- GeForce GT 120
- GeForce GT 130
- GeForce GTX 285
- GeForce 8800 GT
- GeForce 8800 GS
- Quadro FX 4800
- Quadro FX 5600

AMD Qualified GPU Cards

- GeForce 9600M GT
- Radeon HD 4670
- Radeon HD 4850
- Radeon HD 4870
- Radeon HD 5670
- Radeon HD 5750
- Radeon HD 5770
- Radeon HD 5870

Presently. NVIDIA Quadro 4000 is not listed as a supported card for OpenCL under Mac OS X 10.6 Snow Leopard. This may likely change with a future update or the release of Mac OS X 10.7 Lion.

Recent versions of Mac OS X 10.6 Snow Leopard and developer builds of Mac OS X 10.7 Lion have also included preliminary support for OpenCL-qualified cards not listed above. Its pretty likely that the list of officially supported cards will expand to include more in the future.

Questions about SSD and Thunderbolt

We still see common misperceptions posted on forums about associated technology in present and future Mac systems such as SSD and Thunderbolt.

Here is one of the most common questions:

“Will installing an SSD increase my system’s performance (i.e. rendering, processing, etc...)”

That answer is generally no--with a rare exception. Installing an SSD boot drive inside your Mac will make it boot quickly and launch applications fast. The general housekeeping data required by your Mac to do mundane things like run the operating system can load very rapidly onto and off of the SSD, but while it will make your system feel very snappy--it cannot directly speed up processing.

Processing is a function of the CPU (and going forward with OpenCL, the GPU) and having a fast storage device--even an SSD--won’t speed up the CPU’s ability to process data any faster. Unless you are a skilled overclocker, there is really nothing you can do to accelerate the processors in your Mac system by installing a faster boot drive.

There is an exception to this rule. If you are exporting large data sets (such as TIFF or DPX files) then you can accelerate the process by having a large, fast array of drives available to Final Cut Pro or your other processing app. With large files, the bottleneck can change from your system’s processor to your system’s capability to handle thousands of large files that need to be stored somewhere. If your array can’t keep up, then the processing of the exported files needs to be slowed down to accommodate the array’s speed. If you can throw two or three SSD’s in a RAID at this task, then your app is free to churn out these large files to the max of its processing capability.

That being said, we are huge advocates of SSD. Why? Because there are plenty of disk-based tasks that are significantly sped up by changing from a platter-based drive to a solid-state drive. SSD essentially works at the speed of RAM (they are large collections of NAND-based flash memory) so even the slowest SSD generally outpaces the fastest spinning disk drives. Tasks like accessing scratch disks and loading project files are an order of magnitude faster. The appearance of the “beach ball” (a time-out icon displayed while the system shuttles data back and forth from disk to RAM) is nearly eliminated altogether.

A new technology that Apple just released in new Mac systems is called Thunderbolt. This new port is essentially a Mini DisplayPort married up with a 10 Gb/s data stream capability. Thunderbolt works off the PCI Express functionality of the motherboard and provides an extremely easy way to add displays, storage arrays, and other PCIe peripherals. These devices can be daisy chained together similar to how Firewire also connects via a chain-methodology. However, the throughput of Thunderbolt is massive, even compared to the new USB 3.0 standard.

The question we get: *"Will Thunderbolt add any acceleration to my system?"*

The answer to this is similar to the SSD answer: by itself, Thunderbolt does not increase the systems core processing speed. However, Thunderbolt will eventually allow Final Cut Pro systems the capability of easily adding specialized processor cards that can offload some of the heavy lifting that the CPU's are normally required to do.

An example of this is the new Echo box by Sonnet Technology:

http://www.sonnettech.com/news/nab2011

This new product is an external card chassis that can house PCIe cards and connects to the system via Thunderbolt. Cards such as RED Digital Cinema's RED Rocket Card and Matrox's CompressHD cards can be installed in these external chassis. These cards allow processing of R3D and h.264 files to take place on the cards--and not on the CPU. Connecting these cards to Thunderbolt-equipped Macs will free up significant CPU resources to focus on other tasks.

Desktop or Laptop?

Which machine should you get? Your workflow may dictate an answer to this. If you are heavily dependent on traveling and moving quickly, you might need to get a laptop for the sheer sake of size.

However, if you aren't limited--you should really consider getting a Mac Pro. These tower systems from Apple are the most powerful desktop Mac systems available and will offer the most "bang for the buck" in terms of value, processing, and future expandability. Take a look at the chart below to see how each machine in Apple's lineup stacks up.

FEATURE	MAC PRO	MACBOOK PRO	IMAC	MACBOOK	MACBOOK AIR	MAC MINI
Expand RAM up to 64 GB	✔	✘	✘	✘	✘	✘
Hyper Threading	✔	✔	✘ ✔	✘	✘	✘
Turbo Boost	✔	✔	✔	✘	✘	✘
Ability to upgrade GPU	✔	✘	✘	✘	✘	✘
Add up to 12.0 TB of internal storage	✔	✘	✘	✘	✘	✘
Add PCIe 2.0 cards	✔	✘	✘	✘	✘	✘

Presently, certain iMacs (May 2011) have Hyper Threading (Core i7) while others do not (Core i5).

Appendix: External References

Grand Central Dispatch

Apple Technology Brief: Grand Central Dispatch [PDF]

OpenCL

Apple Technology Brief: OpenCL [PDF]

Thunderbolt

Intel Technology Brief: Thunderbolt Technology [PDF]