Choosing a Computer for Running SLX, P3D, and P5

This paper is based on my experience purchasing a new laptop in January, 2010. I'll lead you through my selection criteria and point you to some on-line resources.

32-Bit or 64-Bit?

Most new PCs and laptops have CPUs that can run in 64-bit mode, so the choice of 32-or 64-bit mode is really the choice of an operating system, rather than hardware. Even if you decide to go with a 32-bit operating system, it's still nice to preserve the option of upgrading to 64-bit mode later on. Don't buy a machine with a 32-bit CPU.

The principal advantage of 64-bit mode is being able to address more data. When operating in 32-bit mode, the maximum amount of data that can be addressed is 2GB, or with a little work, around 3GB. For some very large programs, 2-3GB may not be enough memory, so 64-bit mode is the only option.

Wolverine has offered a 64-bit version of SLX for around three years. P3D and P5 are (and will remain) 32-bit applications.

With one exception, 32-bit programs run just fine on a 64-bit system. The exception is programs that use special devices for which 64-bit device drivers are unavailable. If you're contemplating running in 64-bit mode, and you have an older printer, scanner, or other device you'd like to use, be sure to check the availability of 64-bit drivers before taking the leap.

Most SLX programs run about 6-7% slower in 64-bit mode. This is probably because SLX does so much pointer manipulation, and in 64-bit mode, pointers are 64-bit variables, rather than 32-bit variables. While 64-bit mode offers larger address spaces, your programs will require more memory to run, because pointers are bigger.

In my experience, the only programs that actually run a bit faster in 64-bit mode are programs that perform very complicated floating point computations. 64-bit CPUs use a different floating point architecture that is better suited for complicated computations than the 32-bit legacy architecture.

One thing you won't be able to do is run old 16-bit programs on a 64-bit system. You should also beware that some 32-bit programs come with 16-bit installation software, so while they *could* work on a 64-bit system, you probably won't be able to install them.

I have been running XP-64 for over three years on my primary development system, and it has been a very stable platform. I'm running the 64-bit version of Windows 7 Home Premium on my new laptop.

Windows 7?

Support for Windows 2000 will end in July, 2010. Windows XP (Service Pack 3) is now in "Extended Support," which will end in 2014. See the following link for an explanation of Microsoft's support policies:

http://support.microsoft.com/lifecycle/?LN=en-us&x=14&y=5

I'm running Vista on exactly one machine, and that's one too many. In my opinion, Vista is a loser, because it's so annoying to use. It constantly second guesses you and prompts you for approval each time you perform a task it considers to be a potential security threat, e.g., performing a virus scan. Windows 7 has been tamed down in this respect. If you're currently running XP or an even older version of Windows, both Vista and Windows 7 will take some getting used to. In my opinion, if you're going to endure the pain of switching from an older system, go directly to Windows 7.

How Many Cores?

First, I recommend buying a dual-core machine. All but the cheapest machines are dual-core these days. If you haven't used a dual-core machine, let me tell you that they're pretty nice. You can start a long-running application and still have nearly a "second machine" to perform other tasks such as checking your email. Quad-core machines are available, of course, but they're still a bit expensive, and you'll pay more for a quad-core machine than a dual-core machine running at the same clock rate.

Most software doesn't take advantage of multi-core hardware; however, P5 and P3D are notable exceptions. P5 does most of its rendering in the CPU, making very limited use of graphics hardware, so spreading the workload across two CPU cores is very helpful. While P3D makes much greater use of graphics hardware, it uses DirectX 9 to access the hardware, and DirectX 9's capabilities for running multiple execution threads leave something to be desired. DirectX 11 can better exploit multiple cores, but it is available only under Windows 7. Consequently, it will be a while before Wolverine offers a DirectX 11-based implementation of P3D.

When P5 and P3D are run in DLL (library mode), they confine their use to a single core, leaving other cores, if any, for use by the application driving an animation. Concurrent execution of simulation and animation works very well on dual-core machines.

CPU Clock Rate

You should purchase the fastest machine you can afford. Generally speaking, the best bang-for-the-buck can be obtained by buying machines whose CPU clock rates are a notch or two slower than the current state-of-the art.

Clock rate alone doesn't tell the entire story behind performance. We're reaching a point where the speed of light is becoming a limiting factor in improving circuit speeds, so many performance improvements are the result of improved hardware parallelism and improved hardware algorithms. For example, a machine that has three instruction pipelines is generally going to be faster than a machine that has only two.

Hardware resources and algorithms differ considerably between vendors, e.g., Intel and AMD. Until recently, AMD had a distinct edge over Intel in performance vs. clock rate. That is, you could reasonably expect that an AMD chip would outperform a somewhat faster Intel chip. In my recent experience, that distinction is no longer a given. I have a 2.2 Ghz Intel laptop and a 2.2 Ghz AMD desktop machine. For most applications, the Intel laptop is a bit faster than the desktop AMD machine. In fairness to AMD, I must point out that the AMD machine is several years old and has a 512K level 2 cache, while the Intel laptop has a 2MB level 2 cache. For any given application, it's impossible to predict in advance whether it'll run faster on one machine or the other.

Cache Size

Most machines have a 4-level storage hierarchy for data manipulated by the CPU. The fastest storage is in machine registers. The next fastest are level 1 cache storage and level 2 cache storage. The slowest is "real" memory. The unit of traffic to and from real memory and caches is a "cache line," which is typically 64 bytes. Reading a 32-bit value directly from memory requires transfer of an entire cache line to the level 1 cache. Writing an isolated 32-bit value to memory requires first reading the cache line in which the data will be stored and then rewriting the cache line. Writing to random memory addresses is much more expensive than writing to multiple addresses in the same cache line.

There's no such concept as "cache utilization." Level 1 and level 2 caches are always 100% utilized. Therefore, the important issues are their sizes and the effectiveness of the hardware algorithms that manage them. The level 2 cache is the "losers' cache." When data is read into the level 1 cache, a cache line currently in the level 1 cache is evicted to the level 2 cache. If the cache line in the level 2 cache to which the evicted level 1 cache line will be written has been previously modified, the level 2 cache line must be written to memory. When the CPU needs to read data, fastest access is from the level 1 cache. If the data is not in the level 1 cache, but is in the level 2 cache, access is slower, but still considerably faster than real memory access. Thus, data that would eventually be evicted from the level 2 cache frequently gets a "reprieve" when everything is running well.

For programs that access large amounts of memory, the size of the caches and the effectiveness of cache algorithms are *very* important. Level 1 caches are currently typically 64K bytes. Level 2 caches can be 512K, 1MB, 2MB, or even higher.

Memory Speed

Obviously, the faster the memory, the faster the performance. Be sure to look at memory speed when evaluating possible purchases. A "cheaper" machine may be cheaper in part because it uses slower memory. In today's market, you should not settle for memory speeds slower than 800 Mhz. Currently, some hardware features memory speeds as high as 1333 Mhz.

Graphics Performance

For most P3D animations, the limiting factor for performance is graphics hardware. There are some classes of animations for which CPU time is the limiting factor, but these are much less common. One example of CPU-limited P3D animations is the movement of long trains in which many cars are attached to each other, and motion spans many path segments and paths. Such animations consume large amounts of CPU time just keeping all the cars in sync.

P5 uses the CPU to do virtually all of its rendering. It uses the graphics hardware only to transfer a screen image bitmap from main memory into graphics memory, and hence, to the screen. The image transfer cost is independent of the complexity of the images rendered. Thus the only characteristic of graphics hardware that matters is the CPU-to-graphics memory transfer rate. P5 is able to exploit dual core CPUs when it renders screen images; however, processing of trace stream commands is inherently serial and, therefore is performed by a single core.

Laptop graphics hardware is generally slower than desktop graphics hardware for a number of reasons. First, space and weight limitations are more severe in laptops than in desktop machines. A typical high-end desktop video card will have an on-board fan to deal with the heat the card generates. In a laptop, there may not be room for such a fan. Thus, when a laptop and a desktop video card use the same chips, the laptop hardware may run at a reduced clock rate, in order to generate less heat. What you'll typically see is a "1234" designation for desktop hardware and something like "1234M," where the "M" stands for "mobile" in the corresponding laptop hardware.

The following general rules apply to selection of laptop video hardware:

1. The range of performance is quite wide.
2. Laptops with very high-end video hardware are very costly. For example, you could easily pay over $2,000 US for a "gamer" laptop. Such laptops are usually relatively heavy, because they need bigger batteries, bigger fans, etc.
3. It *is* possible to make very reasonable compromises in order to keep costs down. For example, the laptop I recently bought was on sale for under $700. Its graphics hardware is 15-20 times faster than the laptop it replaced, which I purchased several years ago for $1,000 US.
4. Before you buy a laptop, compare the video performance of the graphics hardware for several models of laptops. One of the best sources of information is the following website:

   www.notebookcheck.net

This website provides performance benchmark numbers for a wide variety of laptop graphics hardware. The primary criterion for comparing performance is the "3DMark 05" benchmark.

5. The performance of "integrated" graphics hardware is usually unacceptable. This hardware is built into a machine's motherboard. This type of hardware often lacks

dedicated video memory and takes memory away from the CPU. As an example, consider the Intel Graphics Media Accelerator GMA 4500M. The 3DMark 05 benchmark number for this hardware is 965.

6. The ATI/AMD Radeon Mobility HD4330, available on lots of laptops, provides much better capability, with a benchmark number of 6,710.
7. The laptop I bought has Nvidia GeForce GT230M hardware, with a benchmark number of 10,689. This hardware is described as "discrete," meaning that it plugs into a motherboard. It also has 1GB of on-chip, dedicated video memory.
8. Consider the three preceding examples. They illustrate the fact that for commonly available video hardware, the fastest graphics hardware can easily be ten times faster than the slowest. In my case, I could have spent $800-$1,000 US and have gotten video hardware with lesser performance.

DirectX and Shader Versions

Be sure to buy hardware that supports Microsoft Shader Model 3 or later. One of the principle improvements of level 3 over earlier, lower levels is that level 3 provides hardware support for what's called "instancing." Suppose you're doing an air traffic control application that depicts the motion of up to 5,000 airplanes. With level 3 shaders, P3D will provide a single instance of the airplane geometry to the hardware and then provide vectors of "instances" of the geometry. Each instance has its own transformation of the geometry, including location, orientation, and scale. The size of the instance data is comparatively small. Presenting data to the hardware in this manner provides a compact description of what needs to be done and enables the hardware to divide the work into many parallel threads, exploiting a multiplicity of execution units.

As stated above, P5 and P3D use DirectX 9 technology. Other things being equal, if you have the opportunity to buy graphics hardware that is DirectX 10- or DirectX 11-capable, go with the later version, as P3D will probably exploit their new capabilities some day.

Conclusions

I'm pretty happy with my new laptop. At 2.2 Ghz, with dual cores and 4GB of memory, it can run big SLX models at an acceptable pace. Its Nvidia GeForce GT230M graphics hardware runs P5 and P3D animations very nicely. It really pays to be a careful shopper.

Happy shopping!
Jim Henriksen

January, 2010