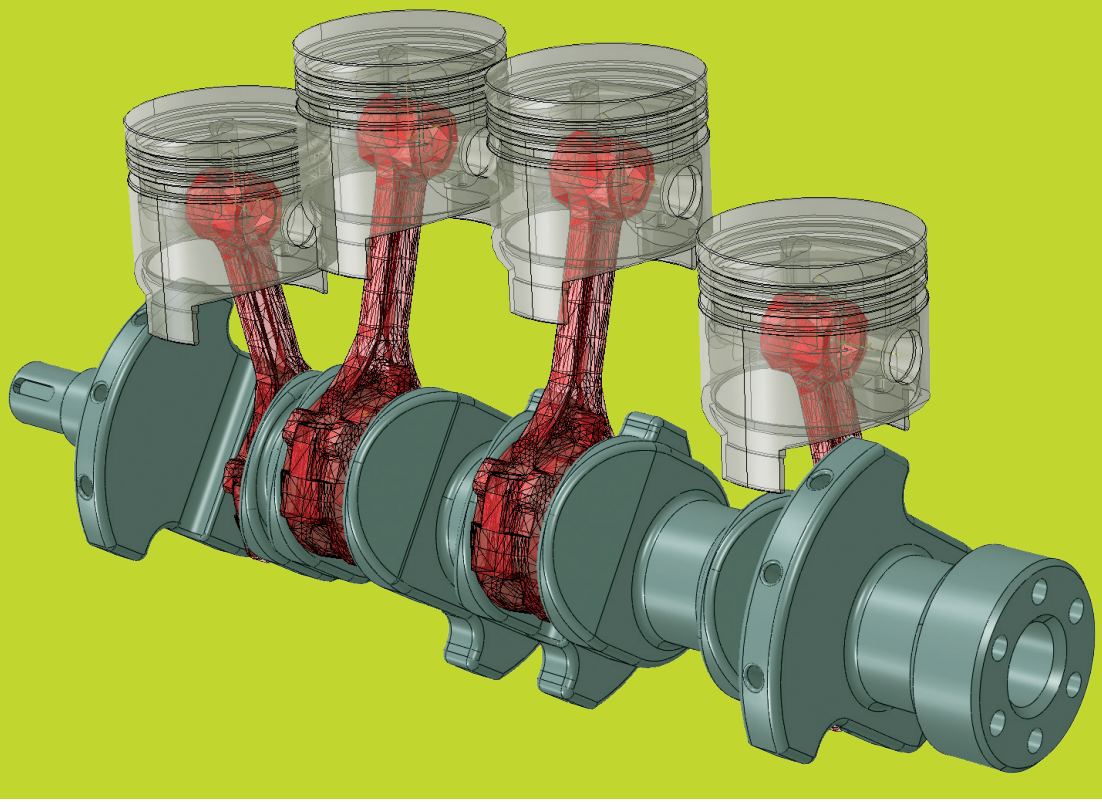
GPU COMPUTE FOR WORKSTATIONS



Model created in Simulia Abaqus: currently one of the only simulation tools to run on AMD and Nvidia GPU hardware

» Thinking about buying a new workstation? It may be time to consider GPU compute. With the ability to accelerate simulation and rendering software, the humble GPU could soon be the most important component inside your machine. **Greg Corke** reports

> **“The performance gains can be huge. Simulation software developers claim a top end GPU can solve a simulation problem two to four times faster than 2 Xeon CPUs”**

The workstation as we know it is changing. The CPU (Central Processing Unit) used to be the only chip that could solve computational problems. Now it has some competition.

The GPU (Graphics Processing Unit) was originally developed for 3D graphics, but can now be used as an extremely powerful co-processor. Its highly parallel architecture, which consists of hundreds of processing cores, makes it incredibly efficient at solving complex operations, such as simulation and ray trace rendering. This is called GPU compute.

GPU compute does not replace the CPU. Simulation or rendering code starts on the CPU and ends on the CPU, but, when appropriate, certain parallel portions get moved to the GPU where they can be calculated faster.

The performance gains can be huge. Leading Computer Aided Engineering (CAE) software developers claim a top end GPU can solve a simulation problem two to four times faster than a pair of multi-core CPUs. Comparisons between ray trace rendering on the CPU and GPU are harder to quantify as the results are more subjective.

In simulation, GPU compute is not just about accelerating solve times. It can deliver more accurate results by increasing the mesh density in larger models. As solve times are reduced it is also possible to explore more alternatives, which can lead to a better design.

A big advantage of GPU compute inside a workstation is having fingertip access to high performance computing, but at the same time it also frees up CPU resources. With its own processing cores and memory, a GPU can perform complex calculations without hogging the CPU and system memory. Conversely, running a ray trace render on a CPU can sometimes make it virtually impossible to perform other tasks such as 3D modelling. It may be a cliché, but GPU compute can be like having two workstations in one.

GPU COMPUTE HARDWARE

For workstations there are two types of GPUs that can be used for compute: those that are that dedicated solely to compute and those can handle both compute and 3D graphics. The

two major manufacturers of these processors are AMD and Nvidia.

High-end GPUs are usually full-length double height PCI Express boards (i.e. they take up two PCI Express slots on the motherboard). They require a spacious ATX workstation chassis and plenty of cooling.

GPU compute boards also require lots of power with some drawing close to 300W. The PCI Express slot can't deliver this much, so additional power needs to be taken direct from the Power Supply Unit (PSU) via a PCIe AUX power connector. PSUs will typically be rated at 1,000W or above.

CUDA VS OPENCL

There are currently two competing programming frameworks for GPU compute: CUDA and OpenCL. Commercial GPU-accelerated software tends to use one or the other, though there are some applications that can use both. It is important to understand the pros and cons of each as this will have a major influence on how you approach GPU compute.

CUDA has been in development since 2004, and is by far the more established of the two frameworks. However, it is a proprietary Nvidia technology and requires Nvidia GPUs, which limits the options when buying hardware. Some CUDA-based applications can also be accelerated by CPUs, but the performance is usually nowhere near as fast. CUDA is not compatible with AMD GPUs.

OpenCL is currently on version 1.2 and is not as mature as CUDA. However, it has the advantage of being an open standard that can execute on all types of GPUs, CPUs and other processors. The importance of this could grow as more applications run on multiple devices, such as tablets or mobile workstations.

OpenCL is developed by the Khronos Group, a non-profit organisation whose members include AMD, Nvidia, Intel, Apple, ARM Holdings and many other hardware and software developers. If the name sounds familiar, the Khronos group also develops OpenGL, the 3D graphics API used by most CAD/CAM/CAE applications.

In terms of commercial software, there are currently far fewer OpenCL applications, though this is expected to change in the coming years as OpenCL matures.

Nvidia has invested heavily in CUDA, which has accelerated its adoption. Its engineering team has helped a number of simulation and rendering software developers implement GPU compute within their applications.

Of the major simulation software developers, DS Simulia, MSC Software, and Ansys all offer CUDA-accelerated CAE applications. In contrast, DS Simulia is currently the only one to have a commercial OpenCL-accelerated offering. That said the other developers are all working on OpenCL applications, with some already in beta.

There's a similar picture in ray trace rendering. Bunkspeed Shot, Catia Live Rendering, RTT DeltaGen and 3ds Max Design are all powered by iRay, a CUDA-accelerated rendering technology that is owned by Nvidia, through its subsidiary, mental images. For OpenCL, the choice is not as big, the two notables being VRAY RT and Optis THEIA-RT, a high-end optical and lighting simulation tool.

While Nvidia publicly backs OpenCL, it obviously has a vested interest in CUDA becoming the industry-standard, primarily so it can sell more of its hardware. AMD fully supports the open standard, OpenCL.

GPU COMPUTE PERFORMANCE

When buying a GPU for compute it's important to understand the key specifications. Modern GPUs support two types of floating point operations: single precision and double precision.

Single precision operations are usually used in ray trace rendering applications. Double precision operations are usually used in simulation (CAE) solvers.

Most modern GPUs offer good single precision performance. However, not all GPUs are optimised for double precision code. This is particularly true of low-end to mid-range GPUs. Interestingly, it is also the case for Nvidia's new high-end Kepler-based Quadro K5000 graphics card. See chart below.

The floating point performance of a GPU is measured in FLOPS (floating-point operations per second), just like it is for a CPU. The latest GPUs offer a TerraFLOP of performance. Matt Dunbar, chief architect, Simulia, equates this to 'tens of x86 servers'. However, there are issues in harnessing all of this power and Dunbar explains that GPUs are not the easiest things to parallelise, or program for, particularly with an existing code base.

FLOPS is not a definitive way of comparing GPU to GPU, but it does give a good indication of what a GPU is capable of. There are other considerations to take into account, such as drivers and how efficiently each software application can use the available hardware.

GPU COMPUTE MEMORY

On board memory is another important consideration for GPU compute. Simulation and rendering problems can be huge and can put a big load on GPU memory. Particularly large problems will not even be able to run.

Unfortunately, there is no way to increase capacity as unlike CPU memory, GPU memory is permanently fixed on the board.

Workstation-class GPU compute capable boards start at 2GB, and currently go up to 6GB. Larger capacities aren't yet available due

GPU CARDS

GPU	AMD FirePro W5000	AMD FirePro W7000	AMD FirePro W8000	AMD FirePro W9000	Nvidia Quadro 5000	Nvidia Tesla C2075	Nvidia Quadro K5000	Nvidia Tesla K20
Graphics / compute	Both	Both	Both	Both	Both	Compute	Both	Compute
Memory (GDDR5)	2GB (NON ECC)	4GB (NON ECC)	4GB ECC	6GB ECC	2.5GB ECC	6GB ECC	4GB ECC	TBA
Memory bandwidth (GB/sec)	102	154	176	264	120	148	173	TBA
Architecture	Southern Islands	Southern Islands	Southern Islands	Southern Islands	1st generation Maximus (Fermi)	1st generation Maximus (Fermi)	2nd generation Maximus (Kepler)	2nd generation Maximus (Kepler)
Single Precision Performance (GFLOPs)	1,300	2,400	3,230	4,000	718	1,030	2,150	TBA
Double Precision Performance (GFLOPs)	79	152	806	1,000	359	515	90	TBA
Price	£379	£569	£999	£2,499	£1,240	£1,750	TBA	TBA

RAY TRACE RENDERING WITH NVIDIA MAXIMUS

Nvidia Maximus supports a number of design visualisation applications, including Bunkspeed Shot and 3ds Max Design 2013.

Both of these tools use iRay, the CUDA-based ray trace renderer from mental images, which is accelerated by Nvidia GPUs. In Bunkspeed Shot it is the default renderer. In 3ds Max Design it is an option alongside the CPU-based renderer, mental ray.

Nvidia Maximus is all about being able to 'design and render' at the same time so we put this to the test by rendering a scene in each application in turn, at the same time as running an interactive 3D graphics test inside SolidWorks.

Bunkspeed Shot has three rendering modes: GPU, CPU and hybrid (GPU & CPU). By default the software uses all of the available GPU resources, but the Nvidia Maximus driver can control which GPU hardware is used (Tesla only or both Tesla and Quadro). This choice has to be made when Bunkspeed Shot is not running, which is a shame as there will likely be times when users want to change settings throughout the day, perhaps during a lunch break in order to knock out a render as quick as possible. As it stands users will need to close down Bunkspeed Shot, change the settings, and reload the application and data.

We tested with all three modes, firstly to see how all the different processors affect render speeds and then to see how they impact 3D performance in SolidWorks.

Whenever the Quadro GPU was used by Bunkspeed Shot for compute the 3D performance in SolidWorks dropped considerably. Model rotation was jerky and erratic and the workstation was pretty unusable.

When the settings were changed to 'Tesla only', the 3D performance in SolidWorks was excellent. There was literally no slow down in frame rates.

It was interesting to note how little the eight core Xeon CPU contributed towards reducing the render time. This demonstrates the importance of having a high-end optimised CUDA GPU for Bunkspeed Shot.

Autodesk's implementation of iRay in 3ds Max Design 2013 is much more flexible. From within the software, users are not only able to control which GPUs are used, but specific CPU cores can also be assigned. In the case of the Intel Xeon E5 2670 the number goes from 0-16 (8 physical cores and 8 virtual hyperthreading cores)

Adding this level of control inside the software gives users much more flexibility in how to distribute workstation resources among various applications. We were able to use 8 CPU cores with little slow down in overall system performance when running other applications.

Arguably, this level of control over CPUs could be attained by setting processor affinity in Microsoft's Process Explorer (**tinyurl.com/D3Dexplorer**), but it's nice to have everything accessible in one place.

We decided to throw the kitchen sink at the system, exploring how the workstation would react when running two rendering processes at the same time: a GPU based iRay render in Bunkspeed Shot and a CPU based mental ray render in 3ds Max Design.

Both processes absolutely hammer all of the available GPU and CPU resources, so we expected some drop in performance, particularly as Bunkspeed Shot uses the CPU to prepare the render. There was some slow down and results were a little erratic at times but, at most, the drop was around 15% in both applications. We were quite impressed with this.

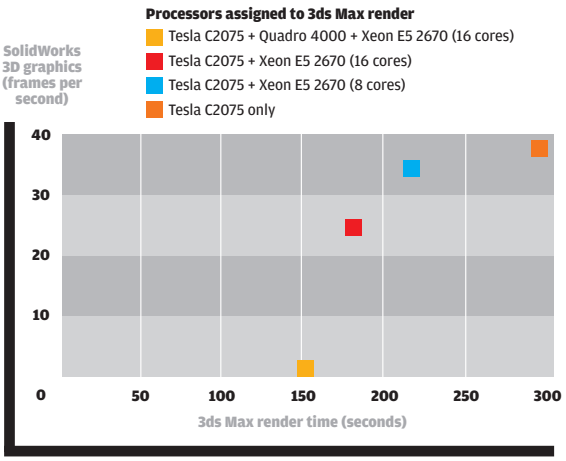
Overall, our tests confirmed just how good Nvidia Maximus is at handling parallel design and visualisation workflows. We expect design/simulation workflows would yield similar results, though there would likely be a bit of slow down due to more frequent interaction between CPU and GPU. Being able to control exactly which GPU resources are assigned to the GPU compute task is a powerful feature of the Maximus driver. However, ideally, this should all be able to be done within the ray trace or simulation tool.

Test machine
Workstation Specialists WS1850 Nvidia Maximus certified 3D workstation.

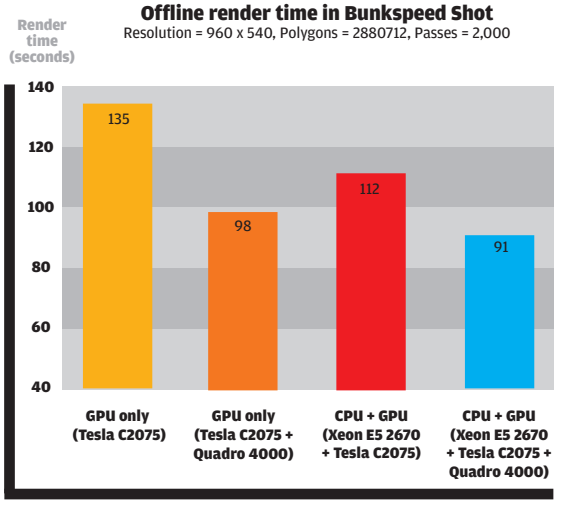
Intel Xeon E5 2670 CPU (eight core) • Nvidia Quadro 4000 GPU (2GB) (graphics) • Nvidia Tesla C2075 GPU (6GB) (compute) • 32GB RAM • 120GB SSD • 2TB SATA.
£4,995

See page 59 for a full review

3ds Max render time vs SolidWorks 3D graphics performance



Offline render time in Bunkspeed Shot
Resolution = 960 x 540, Polygons = 2880712, Passes = 2,000



to hardware limitations. Memory capacity challenges can be compounded if a GPU is being used for graphics and compute at the same time.

There are two types of memory used in GPUs — Error Correcting Code (ECC) memory and non-ECC. Many simulation solvers require ECC memory, which tends to be used in high-end GPUs and not low-end GPUs.

Memory bandwidth is also important, as an incredible amount of data needs to move through the processor. This can be a major limiting factor in general GPU compute performance, more so than the GPU itself.

NVIDIA MAXIMUS

Nvidia has a well-defined approach for GPU compute in workstations, which it calls Nvidia Maximus. An Nvidia Maximus workstation usually has two GPUs: one dedicated to graphics (Nvidia Quadro) and one dedicated to compute (Nvidia Tesla). Some Maximus workstations also feature multiple Tesla cards.

A small but very significant part of Nvidia Maximus is its driver, which gives users full control over which GPUs are used for compute and which are not. This is important as it helps ensure the Quadro GPU is not used for compute by mistake which can significantly slow down 3D performance. Ideally, the user should be able to control GPU usage inside the simulation or rendering software, but this is not always the case.

'First generation' Maximus was based on Nvidia's 'Fermi' architecture and systems have been shipping since late 2011. Nvidia recently announced 'second generation' Maximus, which is based on its 'Kepler' architecture, but products are not yet commercially available.

To date, Nvidia has pre-announced two second generation Maximus products. The Kepler-based Quadro K5000 graphics card (4GB ECC) is due to ship in October 2012. The Kepler-based Tesla K20 GPU compute board will be out in December 2012, but the specification has not yet been announced. However, it is known that the K20 will be optimised for single and double precision performance and will feature ECC memory, probably 8GB or more.

As a workstation technology, the beauty of Nvidia Maximus is it can ensure GPU resources are not shared. A compute task can run quite happily on the Tesla GPU while the rest of the workstation remains free for other tasks. When configured to do so, there is little to no slow down in performance when running a compute task alongside a more traditional CPU/GPU-based task, such as 3D modelling.

Nvidia Maximus is great if you have heavy compute workflows as it really can allow you to 'design and simulate' or 'design and render' at the same time. See box out above.

Nvidia Maximus config utility panel makes it easy to control which GPUs are used for compute

As Maximus is a CUDA technology it currently has the advantage of supporting the greatest number of applications. CUDA-accelerated rendering software includes Bunkspeed Shot, Catia Live Rendering, RTT DeltaGen and 3ds Max Design. For simulation, there's Ansys Mechanical, DS Simulia Abaqus, MSC Nastran and Marc, Altair HyperWorks, FluiDyna, Matlab and others. Nvidia Maximus will also support GPU compute applications that are accelerated through OpenCL.

The downside of Nvidia Maximus is it's an expensive solution. An entry-level first generation Maximus workstation will set you back the best part of £4,000. This makes GPU compute a hard sell for designers and engineers who are more occasional users of simulation or rendering software.

It is possible to use a Quadro GPU for GPU compute. But with Nvidia Maximus this does not appear to be a priority for Nvidia, particularly with regards to double precision applications. Indeed, at 90 GFLOPs, the new Kepler-based Quadro K5000 has only one quarter of the double precision performance of its predecessor, the Quadro 5000.

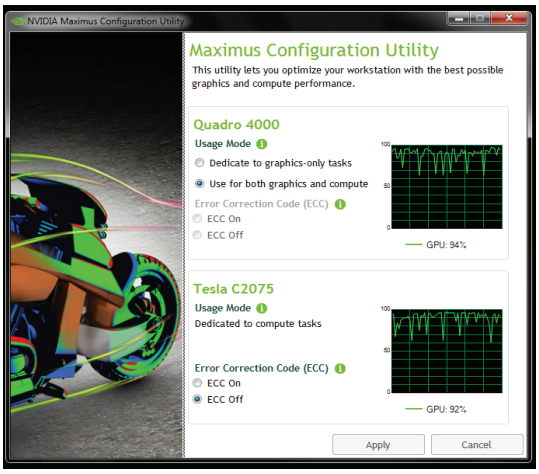
AMD FIREPRO

AMD takes a different approach to Nvidia. First of all it doesn't have dedicated GPU compute boards. Instead it offers a family of general purpose GPUs, the FirePro W Series, which can be used for both graphics and compute. AMD is no longer pursuing its dedicated FireStream GPU compute boards.

Furthermore, in some design and engineering workflows, AMD believes a single FirePro GPU can be used for both graphics and compute without impacting workflow too much. For more GPU intensive workflows, two or more FirePro GPUs can be used.

There are currently four FirePro W Series boards. The FirePro W5000 (2GB) and W7000 (4GB) are best suited to 3D CAD and entry-level ray trace rendering. The FirePro W8000 (4GB ECC) and W9000 (6GB ECC) can handle high-end CAD, simulation, and ray trace rendering.

One big advantage of AMD's single GPU compute solution is cost. Depending on the choice of GPUs, one AMD GPU should be cheaper than two Nvidia GPUs. AMD's solution also offers a lower entry-point to GPU compute, with a particular nod to the FirePro W8000, which boasts impressive single and double precision performance and only costs £999. The FirePro W7000 and W5000 look like interesting propositions for single precision applications, but double precision performance is uninspiring and there is no support for ECC memory.



A downside of AMD's single GPU approach is that applications may well have to fight for GPU resources. And like two kids squabbling over a toy, sometimes no one gets to play.

More often, the loser ends up being 3D graphics performance — to the extent that it can be almost impossible to rotate a model on screen when a GPU compute application is running in the background. This can have a huge impact on design workflow.

This conflict will be particularly true with GPU-based ray trace renderers, most of which use virtually 100% of GPU resources.

It is possible to add additional FirePro GPUs and dedicate one for graphics and one or more for GPU compute. However, AMD does not have the same elegance in its driver as Nvidia does with Maximus, though this could change with future FirePro driver releases. Instead, AMD relies on the ray trace rendering or simulation software developer to provide the means to control which GPUs are used for compute.

In its current form, AMD's single GPU solution looks to be more suited to CAD/simulation workflows. Most simulation software is not able to use all of the available GPU resources, which should leave some free for interactive graphics.

In some CAD applications, such as Inventor or SolidWorks, the load put on the GPU for 3D graphics can be quite low. Certain datasets are limited by the speed of the CPU and can use as little as 15-20% of GPU resources, so there can be a good balance between graphics and compute.

Moving forward, to add another level of complexity, the new AMD FirePro W Series has been designed to be able to run two compute threads and one graphics thread on a single card. Getting three 'kids' to play together nicely will be an even bigger test for AMD's development team. This functionality will be exposed in a driver release later this year.

A big challenge for AMD will be education – how to explain to designers and engineers when to use a single FirePro GPU and when to use more than one, particularly as all workflows are different.

In addition, AMD will need to do a job in terms of marketing. The AMD FirePro brand has traditionally been marketed as 'professional graphics' so some users may find the concept of using a FirePro card for GPU compute confusing. However, AMD is in the process of rebranding to AMD FirePro Technologies, which will help.

In terms of application support for GPU compute, AMD's FirePro cards rely on

MULTITASKING WITH A SINGLE AMD FIREPRO

While AMD can support more than one GPU, in some workflows it believes a single AMD FirePro W series card is fine for both graphics and compute. We tested how this would work in practice running an OpenCL-based simulation task alongside our 3D SolidWorks benchmark with an AMD FirePro W9000 GPU (6GB).

For real world results we would have liked to have used DS Simulia Abaqus, but obtaining a software license was a challenge. Instead we used the SPH Fluid Simulation test from CLbenchmark (*clbenchmark.com*), an OpenCL 1.1 benchmark developed by Kishonti Informatics.

Our first round of testing showed a significant slow down in 3D graphics performance. The SPH Fluid Simulation test grabbed the majority of GPU resources, and the SolidWorks model struggled to rotate on screen.

To help understand exactly what was going on inside the GPU we turned to Microsoft's Process Explorer (*tinyurl.com/D3Dexplorer*).

The SPH Fluid Simulation test used up 85% of GPU resources. In real world CAE software this percentage is likely to fluctuate throughout the simulation by a fair amount as the CPU offloads different tasks to the GPU.

SolidWorks, with our 3D test model, used close to 100% of GPU resources. However, in many 3D CAD workflows, it is unusual to use this much, particularly when using high-end GPUs. Indeed, we loaded up three other SolidWorks assemblies and they took between 15%-25%. The reason for this variance is explained in more detail in a previous article on SolidWorks graphics performance. (*develop3d.com/hardware/solidworks-graphics*)

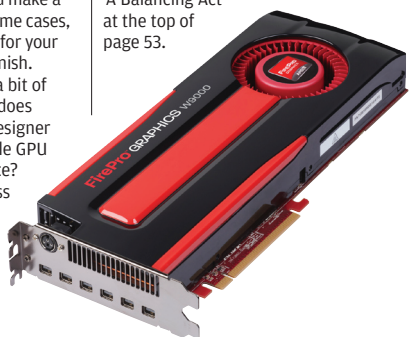
With the new SolidWorks models, our frame rates only dropped by 20-40% when running the simulation test. This is a significant reduction in performance but, importantly, the machine was still responsive and usable.

CLbenchmark also has a ray trace test, which uses close to 100% of GPU resources. When running this alongside our 3D SolidWorks benchmark, model rotation was very slow and erratic, regardless of which model was used.

All of this showed that for some CAD / simulation workflows it is possible to use a single GPU. Depending on the GPU demands of each process users can design and simulate at the same time without crippling workflow. However, when the demands of both processes are high, working with a 3D model that will not rotate exactly when and where you want it to is not practical. You may as well go and make a cup of tea or, in some cases, go home and wait for your compute task to finish.

This throws up a bit of a challenge. How does the engineer or designer know when a single GPU solution will suffice? Microsoft's Process Explorer can help here by assessing the GPU usage of each process, but there's still a bit of trial and error.

Looking to the future, it would make sense to provide control over GPU resources in the FirePro driver – much like setting the affinity of a CPU in Windows Task Manager. We discuss this in more detail in 'A Balancing Act' at the top of page 53.



Bunkspeed Shot: rendering with Nvidia Tesla C2075 and Nvidia Quadro 4000 (no CPU)

OpenCL. As this is quite a new programming framework, commercial simulation and visualisation software is still quite thin on the ground. For visualisation there is Optis Theia-RT and VRAY-RT. For simulation, there is Simulia Abaqus, Autodesk Moldflow and DEM Solutions EDEM.

However, virtually all of the major simulation software vendors are currently developing OpenCL-based solutions, so application support shouldn't be as big a barrier in the future.

A BALANCING ACT

With many CPU-based ray trace renderers, as soon as you hit the render button, you may as well forget using your workstation for anything else. By default, ray trace rendering uses every processing core at its disposal and makes it hard to run other processes efficiently.

One way to stop this crippling your productivity is to set the affinity of your processors. Any single process that runs on Windows can be assigned to specific CPU cores and given a priority. This can all be controlled through Windows Task Manager or Microsoft Process Explorer (**tinyurl.com/D3Dexplorer**). Sometimes this can be done within the rendering software.

Theoretically speaking, we understand this same approach could be applied to GPUs. Users could assign certain GPU cores to 'graphics' and others to 'compute'. This would be particularly beneficial for AMD's single GPU solution, but Nvidia could also benefit by optimising its Quadro in the same way.

While applications that shared GPU resources would likely slow down a bit, it would help ensure no single application grinds to a halt, reducing the impact on workflow. Memory bandwidth could still be a challenge though.

GOING MAINSTREAM?

To date, uptake of GPU compute in CAD software has been slow, but we are now starting to see a significant number of developers integrate the technology within their products.

In design and engineering, GPU compute is currently limited to high-end simulation and ray trace rendering. However, this could change soon. It could become mainstream.

When talking about GPU compute, Nvidia frequently refers to Dassault Systèmes SolidWorks as a mainstream CAD developer that is looking into the technology. We wouldn't be surprised if there were some GPU compute capabilities coming soon in SolidWorks, particularly with the growing influence coming from other DS products.

DS Catia, for example, already supports GPU accelerated ray tracing with Catia Live rendering. And the solver in DS Simulia Abaqus is also GPU accelerated.

Autodesk is also investing in GPU compute. 3ds Max and Maya already have GPU compute capabilities for rendering and fluid flow simulation. Autodesk Moldflow was one of the first commercial CAE software tools to support GPU compute.

However, Autodesk is also investing heavily in the cloud with its Autodesk 360 rendering and Autodesk Simulation 360 services.

CONCLUSION

GPU compute looks set to play a major role in design and engineering in the coming years, particularly in relation to simulation. Performance is everything and being able to slash solve time by two or three times is a massive incentive. Freeing up the workstation's CPU for other tasks can also change product development workflows, enabling users to render and simulate throughout the day, whereas it used to have to happen overnight.

Anyone looking to adopt GPU compute has big choices to make in terms of hardware, weighing up performance (compute and graphics), workflow and cost. These are all complex metrics, but for those looking to take advantage now it may come down to one thing — application availability.

Of course, there are many alternatives for GPU compute. Tasks can also be offloaded to powerful multi-node clusters or sent to the cloud, which offers 'pay as you go' flexibility. And let's not forget the humble workstation CPU, which continues to be popular with many design visualisation software developers.

But GPU compute is all about putting more processing power inside a workstation so it can be at the fingertips of the designer or engineer. This is something that's not been lost on Intel, who is due to launch its own massively parallel PCI Express co-processor later this year. But instead of a GPU, the Intel Xeon Phi is a collection of hundreds of x86-processors based on Intel's Many Integrated Cores (MIC) architecture. So just as the simulation and rendering guys were coming to terms with GPU compute, there'll be another technology to contend with.

amd.com/firepro | nvidia.com/maximus

GPU COMPUTE FOR SIMULATION

Simulation has arguably seen the biggest uptake in GPU compute, but it's taken a long time to get to where it is and is certainly not a mature technology.

In the early days of GPU compute, solver code needed to be completely re-written for the GPU, meaning CAE software developers would have to maintain two code bases. According to Simulia's Matt Dunbar, this was not a viable option. Managing one was hard enough.

A few years ago it became possible to re-compile x86 CPU code for the GPU, so only one code stream needed to be maintained and, hey presto, the applications followed.

All of the major software developers now have certain parts of their solvers that can be accelerated by GPUs. To date, most developments have been for Finite Element Analysis (FEA), including Ansys Mechanical, MSC Software Nastran and Marc, Simulia Abaqus/Standard and Altair HyperWorks.

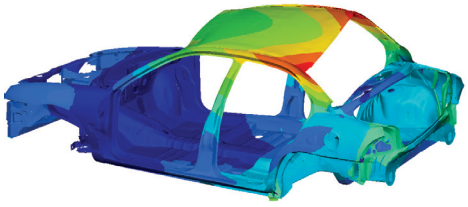
However, there are also GPU-accelerated applications for Computational Fluid Dynamics (CFD) – Altair AcuSolve since 2010, Ansys CFD is currently in beta, LS-DYNA CFD is coming soon and DS Simulia is also optimising Abaqus/CFD code. There are also some specialist CFD software applications that are accelerated specifically for GPUs.

For most CAE software speed-ups are quoted to be in the region of 2x, but some developers claim as much as 4x. This is when comparing the results of a typical dual processor Xeon workstation with and without a single GPU compute board.

DS Simulia says its Abaqus implicit and explicit solvers offer 'very good scalability for complex nonlinear problems.'

Shing Pan, senior director, solver product marketing for Altair Engineering, reports similar benefits with its Radioss Implicit solver "Test results utilising an Nvidia Tesla M2090 GPU [a server-class GPU, slightly more powerful than the Tesla C2075] have shown up to four times faster acceleration in performance, as compared with the latest six-core CPU running the same simulation."

But the performance benefits can be bigger. First generation GPU-accelerated simulation software could only take advantage of one GPU. Now most CAE



developers support two GPU compute boards in a dual processor workstation with lots of memory. However, Simulia says that with Abaqus 6.12 there is only a significant performance gain when the problems reach a significant size.

One of the less publicised benefits of GPU compute in simulation software is the potential cost savings on licensing. Some CAE software developers charge extra to use more processing power.

In the case of Simulia Abaqus, Nvidia explains how 'turning on' the GPU costs a token, which is the same cost as a single CPU core, but offers significantly more performance. For Ansys, it says when users buy the HPC pack to go from 2 to 8 cores, the GPU is included in the price.

IMAGE COURTESY OF ALTAIR