Special Feature

GPUs Do Military General Purpose Computing

Carrier Landing Modeling System Leverages GPGPU Advantages

Well suited for data-intensive, parallel processing, GPGPUs are powerful tools. The technology has been applied to modeling aircraft carrier landing scenarios.

Mark Lovett, Chief Marketing Officer Trenton Technology

S ince Intel introduced the 8088 microprocessor and Motorola released their 68000 over 30 years ago, CPUs have been at the forefront of discussions related to computing power. Users eagerly anticipated each successive iteration that brought with it faster processing speeds and enhanced functionality. But alongside this CPU juggernaut, coprocessors, also known as hardware accelerators, have played a critical role in advancing the ability of computers to deal with increasing volumes of data by offloading the CPU.

In the early days of personal computing Intel's 8087 and Motorola's 68881 math coprocessors were popular with anyone involved in scientific research and analysis, and the subsequent development of graphics coprocessor technology was equally adopted by users of CAD software and gaming machines. More recently, hardware acceleration solutions that utilize Graphics Processing Units (GPUs) working in tandem with Central Processing Units (CPUs) have fostered the GPGPU (General-purpose computing on graphics



processing units) computing paradigm, and this hybrid architecture has proven itself to be an ideal platform for data-intensive military applications. Table 1 lists the kinds of military applications suited for GPGPU technology.

A Powerful Combo

The basic reason this combination works so well is that CPUs and GPUs process instruction and data sets differently, and there is great synergy in utilizing both approaches to address the needs of data-intensive applications. As a general rule, CPUs are designed to excel at serial tasks, processing a range of diverse functions as quickly as possible. But the inherent "general purpose" nature of CPUs, combined with architecture advances such as superscalar design, out-oforder execution, dynamic scheduling, branch prediction and simultaneous multithreading, have served to limit the number of cores that can be packed onto a single die while still remaining within acceptable power and thermal envelopes. This explains why many supercomputers employ hundreds of servers providing thousands of PowerPC, Opteron or Xeon processors clustered by way of an InfiniBand interconnect.

GPUs, on the other hand, are built with hundreds of simpler cores designed as massively parallel vector processors and tailored for high-performance solutions that involve a high degree of data parallelism. While originally targeted at graphicsintensive applications to perform texture mapping and polygon rendering, GPUs happen to be a natural fit for mathematically intensive problems that involve large datasets, taking advantage of this massively multithreaded computing architecture.

Simpler Cores in Parallel

But this shift from serial to parallel processing in hardware must also be mirrored in the writing of code. "Programming has typically been focused on a linear process of executing program functions," says Allan Snavely, associate director for the San Diego Supercomputer Center, "but programs that involve parallel processing of very large data sets outside the CPU require a different mindset—the programmer needs to think about how the data can be broken up into smaller units for simultaneous processing across hundreds or



Figure 2

Shown here is an artist's rendering of an X-47B Carrier-Capable Stealth UAV landing on a carrier deck.

Applications Suited for GPGPU Technology	
~	Video encoding/decoding and stabilization
~	Object detection and pattern recognition
~	Radar, sonar and radio signal processing
~	Still photo and video image enhancement
~	Intelligence, surveillance & reconnaissance
~	Encryption/decryption and cryptanalysis
~	Physics-based modeling and simulation
Table L	

The GPGPU paradigm has proven itself to be an ideal platform for data-intensive military applications.

thousands of cores, and that's a skill most programmers are still learning."

With potentially thousands of cores available, the key to maximizing performance is to keep those cores as busy as possible, and that requires application developers to write or modify their code in a way that extracts computationally intensive units of data staged in the CPU and maps them to run on the GPU while allowing the CPU to handle the remainder of the application. Figure 1 shows the architecture of NVIDA's Fermi GPU.

While GPU hardware is designed to manage thread execution and scheduling, the programmer is still responsible for supplying data to thousands of simultaneous threads. Traditionally, GPGPU used to require a great deal of time and effort in order to map algorithms using





graphics languages, but the release of new programming tools has made life simpler and accelerated the adoption of GPGPU computing. One such solution was Brook, an extension of standard ANSI C that incorporates the concept of data parallel computing, which in turn allows the GPU to function as a streaming coprocessor.

OpenCL (Open Computing Language), initially developed by Apple, is an open standard programming framework that allows code to execute across both CPUs and GPUs. With broad industry acceptance, implementations of OpenCL are available for Intel, AMD/ATI, S3 and NVIDIA hardware. Previous to the release of OpenCL, NVIDIA created CUDA, a proprietary solution allowing programmers to write in high-level languages such as C, C++ and Fortran, as well as utilize OpenCL and DirectCompute APIs.

Keep CPU Cores Busy

Modeling and simulation applications, with their large data sets and reliance on mathematical functions, can take full advantage of GPGPU computing's massively parallel computational model. As a result, more complex problems can be addressed and additional scenarios analyzed. This shift in processing capability is significant, as real-world testing of multiple scenarios significantly increases project cost, and in many cases is virtually impossible as conditions that require testing occur at unpredictable times.

Such is the case when trying to analyze aircraft carrier operations at sea. Setting a jet fighter down on a moving carrier deck is never an easy task, and the degree of difficulty increases during adverse weather. As a rigid body residing within three-dimensional space, a carrier is subject to the six degrees of freedom while at sea, requiring algorithms to deal with displacement motions (heave, sway and surge) as well as angular motions (yaw, pitch and roll) as the ship interacts with oceanic and atmospheric conditions.

As a result, the Navy conducts aircraft missions when conditions are within predetermined operational envelopes, but testing the limits of such operational envelopes using aircraft and pilots is not a practical option due to safety concerns. This is why computer models are used on a range of scenarios to more accurately predict how aircraft and the carrier react under different sets of environmental conditions.

An added benefit to creating computer models is that flight scenarios can be saved and subsequently used with flight simulators to train pilots, giving them the valuable experience of landing in a variety of less than optimal conditions without risking lives or aircraft. Such simulation models not only apply to piloted aircraft, but are also relevant for operating unmanned aerial vehicles off a carrier flight deck (Figure 2).

Modeling Complex Scenarios

To address this complex issue, EM Photonics is developing a hardware-accelerated computational fluid dynamics (CFD) tool based on NVIDIA GPGPU technology that will rapidly and accurately model aircraft interaction with naval vessels, especially with regard to the vessel's airwake. Once developed, these CFD solvers can also be used to model a range of other scenarios where small moving objects interact with larger moving structures, from helicopters landing on ships to deploying payloads from aircraft.

The first challenge for the team involved verification that complex CFD computations map well to the multicore NVIDIA GPU hardware and that significant performance increases were possible by way of utilizing a massively parallel processing architecture. CFD solvers are generally optimized for a particular application of interest, and in this case the objective is the modeling of the Dynamic Interface (DI) for complex CFD problems involving very large and very small moving objects.

While the programming model for GPU computing is relatively new, developers must often deal with code that has been around for a long time. "Creating wholly new CFD solvers based around CUDA is interesting, but not practical because customers are familiar with their present applications and don't want to validate and re-learn new solvers, so the need to modify existing code is a reality" said John Humphrey, Director of Computing at EM Photonics. "What we look for is the roughly 10% of program code that's responsible for consuming 90% of the runtime cycles, as parallelizing the most compute-intensive portions of the program will yield the greatest performance improvement when applying GPGPU technology."

During Phase I of the programwhich was designed to be a proof of concept-compute time of a solver for the Euler equations was reduced from 18 hours to 20 minutes, resulting in a 54x improvement. While this level of performance gain is quite impressive, the challenge for the project's next phase is to deal with computations that can take up to 150,000 CPU hours to run. According to Humphrey, in Phase II they will build on this prototype to create a full solver that will target both desktop users with GPU coprocessors as well as large GPU clusters meant for running the most difficult simulations. In the process they will also encapsulate functionality that can be leveraged across multiple CFD engines.

GPGPU Hardware Solutions

While GPGPU computing is often deployed on personal computers within office environments, or on commercialgrade servers within data centers, rugged military-grade rackmount computers are required to address the harsh realities of shock and vibration, extended temperatures and SWaP considerations. Trenton's TRC4008 (Figure 3) is a 4U rackmount computing solution that includes a 14-slot PCI Express backplane with four doublewide x16 PCIe slots that can support up to four NVIDIA C-Series Tesla GPU cards. These PCIe slots support either PCIe 2.0 or 1.1 electrical interfaces by virtue of the backplane's PCIe Gen 2 switch and the CPUs used on the single board computers.

The TRC4008 system can support either a single-processor or dual-processor single board computer. The SBCs in turn may feature either dual or quadcore CPUs to handle the application's serial computing tasks while the parallel processing is offloaded to the bank of GPUs. The choice of SBC largely depends on the overall application in terms of the amount of serial processing vs. parallel processing needed to produce the most efficient GPGPU system solution. The rugged nature of this integrated GPGPU system, coupled with its inherent application flexibility and long-term configuration stability, provides military and government users with a hardware platform choice designed from the ground up to handle the needs of MIL-COTS field deployments.

Trenton Technology Gainesville, GA. (770) 287-3100. [www.trentontechnology.com].