GigaThread™ Thread Scheduler

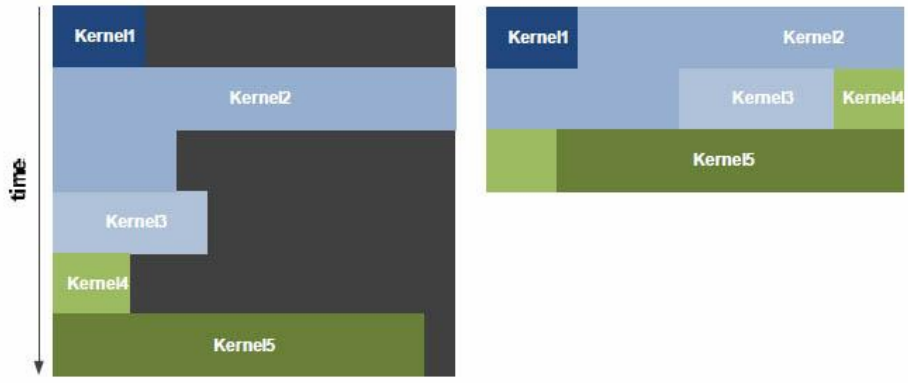
One of the most important technologies of the Fermi architecture is its two-level, distributed thread scheduler. At the chip level, a global work distribution engine schedules thread blocks to various SMs, while at the SM level, each warp scheduler distributes warps of 32 threads to its execution units. The first generation GigaThread engine introduced in G80 managed up to 12,288 threads in realtime. The Fermi architecture improves on this foundation by providing not only greater thread throughput, but dramatically faster context switching, concurrent kernel execution, and improved thread block scheduling.

10x Faster Application Context Switching

Like CPUs, GPUs support multitasking through the use of context switching, where each program receives a time slice of the processor's resources. The Fermi pipeline is optimized to reduce the cost of an application context switch to below 25 microseconds, a significant improvement over last generation GPUs. Besides improved performance, this allows developers to create applications that take greater advantage of frequent kernel-to-kernel communication, such as fine-grained interoperation between graphics and PhysX applications.

Concurrent Kernel Execution

Fermi supports concurrent kernel execution, where different kernels of the same application context can execute on the GPU at the same time. Concurrent kernel execution allows programs that execute a number of small kernels to utilize the whole GPU. For example, a PhysX program may invoke a fluids solver and a rigid body solver which, if executed sequentially, would use only half of the available thread processors. On the Fermi architecture, different kernels of the same CUDA context can execute concurrently, allowing maximum utilization of GPU resources. Kernels from different application contexts can still run sequentially with great efficiency thanks to the improved context switching performance.



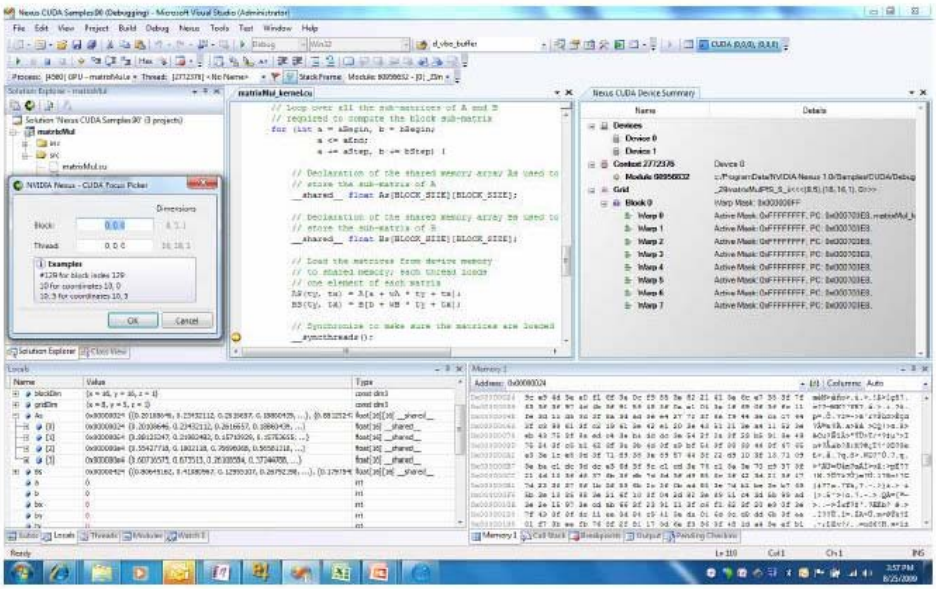
Serial Kernel Execution **Concurrent Kernel Execution**

Introducing NVIDIA Nexus

NVIDIA Nexus is the first development environment designed specifically to support massively parallel CUDA C, OpenCL, and DirectCompute applications. It bridges the productivity gap between CPU and GPU code by bringing parallel-aware hardware source code debugging and performance analysis directly into Microsoft Visual Studio, the most widely used integrated application development environment under Microsoft Windows.

Nexus allows Visual Studio developers to write and debug GPU source code using exactly the same tools and interfaces that are used when writing and debugging CPU code, including source and data breakpoints, and memory inspection. Furthermore, Nexus extends Visual Studio functionality by offering tools to manage massive parallelism, such as the ability to focus and debug on a single thread out of the thousands of threads running parallel, and the ability to simply and efficiently visualize the results computed by all parallel threads.

Nexus is the perfect environment to develop co-processing applications that take advantage of both the CPU and GPU. It captures performance events and information across both processors, and presents the information to the developer on a single correlated timeline. This allows developers to see how their application behaves and performs on the entire system, rather than through a narrow view that is focused on a particular sub-system or processor.



NVIDIA Nexus integrated development environment

Conclusion

For sixteen years, NVIDIA has dedicated itself to building the world's fastest graphics processors. While G80 was a pioneering architecture in GPU computing, and GT200 a major refinement, their designs were nevertheless deeply rooted in the world of graphics. The Fermi architecture represents a new direction for NVIDIA. Far from being merely the successor to GT200, Fermi is the outcome of a radical rethinking of the role, purpose, and capability of the GPU.

Rather than taking the simple route of adding execution units, the Fermi team has tackled some of the toughest problems of GPU computing. The importance of data locality is recognized through Fermi's two level cache hierarchy and its combined load/store memory path. Double precision performance is elevated to supercomputing levels, while atomic operations execute up to twenty times faster. Lastly, Fermi's comprehensive ECC support strongly demonstrates our commitment to the high-performance computing market.

On the software side, the architecture brings forward support for C++, the world's most ubiquitous object-orientated programming language, and Nexus, the world's first integrated development environment designed for massively parallel GPU computing applications.

With its combination of ground breaking performance, functionality, and programmability, the Fermi architecture represents the next revolution in GPU computing.