

Peter Holvenstot
Questions: CUDA 5 and OpenCL

- What is different about the way kernels are executed on devices supporting CUDA 5.0?
 - Kernels are scheduled and dispatched from the card, not the host
- Name one important feature enabled by this change
 - Dynamic Parallelism, Kernel Recursion, Task Parallelism
- What
 -
- When using task parallelism, how can you guarantee concurrency of data?
 - Launch kernels into named streams and check for sync
- If a thread wishes to complete all kernels it has launched, what call should it make and which kernels will be completed?
 - `cudaDeviceSynchronize()`, synchronizes all launches by any kernel in the block
- CUDA 5.0 allows direct memory access to any device on the PCI-E bus. Name one device that might be usefully interfaced by this method.
 - Switched-fabric communications link, such as InfiniBand
 - A problem-specific coprocessor, such as FPGAs, DSPs, etc
- What is the “draw” of OpenCL code versus CUDA or CPU code?
 - The same code will execute across very different devices without rewriting or recompiling.
 - Manufacturers may implement efficient Intermediate Language interpreters for their own device
 - Manufacturers may create (non-portable) extensions to the specification or include “built-in” kernels to provide efficient access to additional device functionality.
- Name three types of devices OpenCL code may be executed upon?
 - CPUs, GPUs, FPGAs, DSPs, etc
- Who implements the compiler for OpenCL kernel code?
 - The manufacturer of the device
- Name a major company involved in OpenCL.
 - Apple, AMD/ATI, Nvidia, Intel
- How may OpenCL applications reach high performance under diverse environments?
 - Use a tuning framework to select optimal kernels or grid topology