C++ AMP, Part 2 of 2: Memory Layout and Support

page 1

Meet the expert: John Stratton, Ph.D., is a senior architect at Multicoreware Inc. and a visiting lecturer at the University of Illinois at Urbana-Champaign. John has been at the forefront of research and education in heterogeneous computing, reaching hundreds of students through the Virtual School of Computational Science and Engineering's courses on heterogeneous computing and optimization for scientific applications. John writes papers and articles for leading academic conferences and journals as well as broad-reaching publications such as IEEE Computer. He is also an active participant and presenter at several industry and technology groups and events across the country.

Prerequisites: This course assumes that you have a good understanding of core C++ concepts, included classes, objects, containers, and iterators. You should also be familiar with Visual Studio 2012 for Visual C++ development, including compilation, testing, and debugging. Although not required or expected, you may get more out of some parts of the course if you are familiar with multithreaded programming, Visual Studio 2012's debugging capabilities for multiple threads, and basic computer architecture concepts.

Runtime: 02:22:47

Course description: In this course you'll learn about how accelerator hardware is designed and integrated into the system. With that foundation, we can start talking about what you can expect from the system when you use various C++AMP features. Specifically, we will talk about data transfers to and from the accelerator, memory layout and memory accesses from the accelerator, and thread execution and control flow on the accelerator. Then we'll cover what support Microsoft's Visual Studio 2012 has for C++ AMP.

Course outline:

Memory Layout Overview
- Introduction
- GPU Architecture Overview
- Minimum Scale of Parallelism
- Demo: Scale and Preformance
- Demo: Benchmark Results
- Summary

Memory Layout and SIMD
- Introduction
- Memory Layout and Accesses
- Good Access Patterns
- Demo: Transpose Operation
- Implicit SIMD Execution
- Divergent Penalties
- Demo: Divergence
- Demo: Divergence Problems
- Summary

Data Transfers
- Introduction
- Host-Accelerator Data Transfers
- When Data Transfers Happen
- Demo: Data-Transfers
- Demo: Array View
- Summary

Windows Support
- Introduction
- C++AMP uses Direct Compute
- Demo: AMP Implementations
- Demo: Multiple Accelerators
- Summary

Debugging
- Introduction
- C++AMP Debugging
- Demo: Debugging C++Amp
- Demo: Debugging Tools
- Demo: Freezing Threads
- Debugging Parallel Kernal Code
- Summary

Tiling
- Introduction
- Tiled Extents and Indexes
- Tiled Accelerator Execution
- Demo: Tiled Extents
- Tiled Accelerator Execution (2)
- Demo: Tile Size
- Demo: Tile Variables
- Summary