Dear Prospective Student,

nCore HPC is committed to being the leader in providing premier hands-on, instructor-led courses focused on multicore and manycore technologies. Our expert engineers and instructors have spent countless hours creating and refining the nCore course curriculum, providing you with state-of-the-art solutions for managing today’s changing computing landscape.

nCore’s complete series of training courses not only will help you develop critical skills to conquer parallel programming, but also will help your organization leverage multicore technologies effectively, increase application performance, reduce time-to-market, and enhance your organization’s competitive edge.

If our standard training courses do not satisfy your educational needs, we also offer custom training courses conducted at your facility anywhere in the world. nCore also offers mentoring and organizational implementation of parallel-aware software development tools and methodologies designed to reduce the training-to-implementation gap.

Please explore this course catalog to discover nCore’s complete offerings and determine which training courses are right for you, or visit http://www.ncorehpc.com/training/.

Ian Lintault
Managing Director
nCore Training Courses
Complete hands-on training courses offering state-of-the-art solutions to increase your software’s correctness and performance on multicore and manycore platforms

Introduction

nCore Design is the world leader in providing instructor-led, hands-on intensive training workshops focused on multicore technologies, GPUs, and embedded systems.

Our passion for the art and science of high-performance software provides you with the solid theoretical foundations and experience to jump-start your next parallel project.

What's Covered?

• **Programming multicore processors** – Learn to correctly architect, design, and develop efficient parallel applications for multicore processors by understanding how to deal with synchronization, mutual exclusion, processor caches, shared memory, and special methods for multithreaded programming using modern methods, such as OpenMP and Intel Threading Building Blocks.
• **Mastering advanced multicore techniques** - Learn how to successfully profile and optimize multicore software in order to achieve maximum performance. You will master advanced multicore programming concepts, debugging and profiling multicore programs, and practice parallelizing applications.
• **Programming GPU processors** - Learn the fundamentals of GPU programming through in-depth, hands-on laboratories using CUDA and OpenCL. By integrating multicore software development techniques and mathematical algorithms used by scientists and engineers, students gain expertise in developing high-performance GPU software on Linux or Windows platforms.
• For complete course descriptions, please visit http://www.ncorehpc.com/training/

Why Enroll?

• All courses taught by experts in multicore, multithreaded, GPU/manycore, and parallel software systems and architectures.
• Courses offered worldwide and without the need for travel – on-site classes offered at your convenience at your facility or a nearby partner training center.
• In addition to nCore’s standard courses, comprehensive training workshops and courseware can be customized to meet your organization’s educational needs.
• Optional on-site follow-on consultation to bridge the gap between the course and your organization’s strategic objectives.

What Are The Benefits?

• Avoid the substantial technical risk of not having the necessary skills to tackle the multicore and manycore parallel computing paradigm.
• Comprehensive training workshops offer students an in-depth overview of fundamental concepts, while offering advanced training and practical advice on programming.
• Online training delivery platform and instructor-led, hands-on laboratories provide detailed instruction and deep background, increasing students’ knowledge and skills.
• Gain critical insights on how to improve your software’s performance and efficiency.
• Acquire a competitive advantage by learning to correctly design efficient multicore-aware algorithms necessary for applications to run on modern platforms, while reducing time-to-market and product development slippage.
• Complete set of standard training courses, custom course options, and consulting help fine tune your organization’s capabilities and eliminate the training-to-implementation gap.
How Do I Select The Right Course?

Basic Advanced

NCT-100 Programming Multicore Processors

NCT-200 Advanced Multicore Techniques

NCT-300 Programming GPU Processors

NCT-450 MCAPI Programming

NCT-6XX nCore Custom Training Course

What Do Clients Say?

• “As always, the instructor’s knowledge level and material presented are great. The instructor truly is an expert and it shows in every presentation.”

• “Not only was a great deal of information presented, but having . . . labs ready to run was really useful.”

• “By presenting pthreads, OpenMP, and TBB it was easy to see how they are all really trying to do the same thing for the programmer — just different ways of solving a problem. This helped me take a more structured approached to multi-threading my own work.”

• “Thanks again for a great three days. This was the kick I needed to get moving.”

About nCore

nCore HPC is a global provider of professional services and systems focused on high performance, low latency and scalability in embedded computing. nCore delivers state-of-the-art solutions to government agencies, high-technology organizations, defense, research, biomedical and financial companies.

nCore is a working group member of the Multicore Association.

Course Registration

Contact nCore for more information: info@ncorehpc.com
NCT-100 Programming Multicore Processors

Three-day intensive training course covering all aspects of programming multicore processors using advanced methods and techniques

**Course Overview**

This course covers concepts and approaches for programming multicore processors in C/C++. From recognizing parallelism opportunities to designing multithreaded algorithms, this course teaches students how to deal with the necessary aspects of multithreading, synchronization, multicore processor caches, shared memory, and specialized methods for multithreaded programming using modern methods, such as OpenMP and Intel Threading Building Blocks.

**Course Objectives**

- Review theoretical background covering multicore processor architecture, concurrent programming, and parallel programming concepts and considerations.
- Cover critical concepts such as implicit and explicit parallelism, atomicity, synchronization, shared memory, cache coherency, Amdahl’s Law, Flynn’s processor classifications, and Little’s law in detail.
- Recognize the best parallelism opportunities and explain the advantages of using threads to obtain concurrency using various analysis techniques, compositional approaches, and parallel design patterns.
- Learn how to avoid synchronization pitfalls such as starvation, deadlock, live lock, and data races.
- Learn how to implement and tune parallel algorithms.
- Define and use different synchronization methods effectively, including mutexes/critical sections and condition variables.
- Explain operating system interactions and the relationship between shared memory and threads.
- Explain what aspects of the operating system affect programming, how to deal with shared memory effectively, CPU selection, CPU-specific binding of threads, thread specific data, and kernel-level scheduling.
- Understand and use threads with specific technologies and programming methods, such as the Windows API, POSIX pthreads, Intel TBB, and OpenMP using C/C++ and the Intel Compiler.
- Gain hands-on experience with the Intel Compiler to build and run multithreaded programs during the laboratories.
- Learn best practices to deal with MT-unsafe libraries and how to write new thread-safe libraries.

**Benefits**

- **A comprehensive training workshop:** This course offers an in-depth overview of fundamental concepts, while offering advanced training and practical advice on C/C++ programming of multicore processors using modern methods.
- **Gain critical insights on how to improve your software’s performance:** This course is designed to give you key skills using specialized tools to help you to correctly architect, design, and develop efficient parallel applications for multicore processors.
- **Additional hands-on learning:** This course provides laboratory sessions in writing multithreaded programs and exercises on practical parallelization of legacy software. It also includes walk-through laboratory exercises designed to increase your understanding of multithreading.

**Who should attend**

Software architects, software developers, software team leaders and managers seeking to understand and implement efficient software running on multicore processors. Knowledge of the C++ programming language and C++ software development experience is a pre-requisite for this course.

**About nCore**

nCore HPC is a global provider of professional services and systems focused on high performance, low latency and scalability in embedded computing. nCore delivers state-of-the-art solutions to government agencies, high-technology organizations, defense, research, biomedical and financial companies.

nCore is a working group member of the Multicore Association.
NCT-200 Advanced Multicore Techniques

Three-day intensive training course focused on profiling and optimizing multicore microprocessor software

Course Overview

This course covers concepts and approaches related to developing, profiling, tuning, and optimizing parallel software on multicore platforms from Intel, AMD, and Oracle Sun. Critical concepts and applied techniques are covered in detail to help you extract maximum performance from your applications. Specific techniques for tuning NUMA architectures, data race detection, profiling, and debugging are taught along with hands-on experience using Intel Threading Building Blocks and Array Building Blocks to parallelize software.

Benefits

- **A comprehensive training workshop:** This course offers an in-depth overview of fundamental concepts, while offering advanced training and practical advice on profiling and optimizing C/C++ programs on multicore microprocessors.
- **Gain critical insights on how to improve your software’s performance:** This course is designed to give you key skills using specialized tools to help you correctly create, optimize, and tune parallel applications for multicore processors.
- **Additional hands-on learning:** This course provides laboratory sessions in optimizing and debugging parallel applications. It also includes walk-through laboratory exercises designed to increase your understanding of parallel tools, such as profilers and debuggers.

Course Objectives

- Receive an in-depth theoretical background, covering processor memory models, NUMA hardware, operating systems kernels, multicore tuning, and modern multicore processors from Intel, AMD, and Oracle Sun.
- Cover critical concepts, such as sequential consistency, NUMA architectures, thread and memory affinity, locality, profiling, and tuning.
- Learn how to profile and tune parallel algorithms for best performance on multicore hardware.
- Define and correct multicore problems, such as false sharing, data races, unnecessary dependencies, load balancing, poor locality, and numerical performance.
- Explain operating system interactions and the relationship between shared memory and threads, including information on NUMA kernel support and multicore and power scheduling on Linux and Solaris operating systems.
- Explain how to deal with shared memory effectively and scalably including CPU selection, CPU-specific binding of threads, thread specific data, lock optimization, cache blocking, first-touch placement and data locality.
- Understand and use parallel technologies and programming methods, such as Intel TBB and Intel ABB using C++ and the Intel Compiler to express parallelism.
- Find data races using the Intel Thread Checker and Valgrind’s ThreadSanitizer. Introduce pintool for dynamically instrumenting programs.
- Learn to use TAU (Tuning and Analysis Utilities), OpenSpeedShop, and likwid to profile applications.
- Learn to use Allinea DDT for debugging and visualizing parallel software.
- Gain hands-on experience with the Intel Compiler to build, tune, and run multithreaded programs during the laboratories and case studies.

Who should attend

Software architects, developers, team leaders, and managers seeking to optimize and tune software running on multicore processors. Knowledge of parallel software development, the C++ programming language, and intermediate C++ software development experience is a pre-requisite for this course.

About nCore

nCore HPC is a global provider of professional services and systems focused on high performance, low latency and scalability in embedded computing. nCore delivers state-of-the-art solutions to government agencies, high-technology organizations, defense, research, biomedical and financial companies.

nCore is a working group member of the Multicore Association.

Course Registration

Length: 3 Days    Cost: $3495

By E-mail: training@ncorehpc.com

nCore may make changes to specifications & product descriptions at any time, without notice. All trademarks & copyrights are the property of the respective owners. Copyright (C) 2008-2015 nCore HPC LLC.
NCT-300 Programming GPU Processors

Four-day intensive training course covering all aspects of programming GPU processors using advanced methods and techniques

**Course Overview**

This course covers concepts and approaches related to programming GPU processors using both CUDA and OpenCL. Extensive coverage of GPU hardware, memories, data transport, and performance optimization enable the student to understand the fundamental aspects of GPU programming. In-depth, hands-on laboratories demonstrate how to apply common numerical methods to GPU processors using both the native APIs and open source numerical libraries. This course also covers methods of integrating the Intel TBB threading abstraction layer with GPU software APIs.

**Course Objectives**

- Install GPU libraries and drivers, and compile CUDA/OpenCL programs on Linux and Windows operating systems.
- Understand nVidia GPU hardware and the underlying technical concepts, including SIMD processing and hardware threading architectures.
- Understand the different GPU programming APIs and their appropriate use with various applications.
- Learn single and double precision floating point calculations.
- Recognize the difference between GPU memory types and the advantages and disadvantages of each.
- Effectively orchestrate the transport of data to and from GPU memory.
- Correctly implement two common types of numerical algorithms - Matrix Multiplication and Reduction.
- Cover performance optimization, including the cudaprof profiling tool, loop unrolling, coalesced memory access, memory bandwidth estimations, and occupancy.
- Participate in labs for commonly used open source computational libraries - CUBLAS, CUFFT, and CUDPP.
- Learn to meld multicore processors and GPUs to take maximum advantage of modern platform performance.
- Learn to integrate the Threading Building Blocks threading abstraction layer with GPU code and migrate TBB primitives to the GPU.
- Discover how to take advantage of multiple GPUs in the same server.
- Cover cudagdb for CUDA debugging, including the use of emulation mode with valgrind.
- Participate in extensive hands-on laboratories with code examples, using both CUDA and OpenCL.

**Benefits**

- Teaches everything necessary to start developing high-performance GPU software on Linux or Windows platforms.
- Covers both CUDA and OpenCL, including open source computational libraries, such as CUBLAS, CUFFT, and CUDPP.
- Explains how to integrate multicore software development techniques with GPUs to increase performance.
- Offers a detailed overview of fundamental concepts, while providing advanced training and practical advice on GPU programming.
- Provides in-depth instruction that increases students knowledge and skills through an online training delivery platform and instructor led, hands-on laboratories.

**Who should attend**

Software architects, developers, team leaders, and managers seeking to develop GPU software. Knowledge of computer architectures and intermediate C++ programming, as well as software development experience are mandatory prerequisites for this course.

**About nCore**

nCore HPC is a global provider of professional services and systems focused on high performance, low latency and scalability in embedded computing. nCore delivers state-of-the-art solutions to government agencies, high-technology organizations, defense, research, biomedical and financial companies.

nCore is a working group member of the Multicore Association.

---

**Course Registration**

Length: 4 Days  Cost: $3495

By E-mail: training@ncorehpc.com
NCT-450 MCAPI Programming
Two-day Multicore Communications API Programming Intensive Training Course

Course Overview
This course covers concepts and approaches related to the design and implementation of multicore software using MCAPI, a standard lightweight communications API created by the Multicore Association and defined for closely distributed multicore embedded systems. MCAPI captures the basic elements of communication and synchronization that are required for such systems, simplifying the migration of applications from a single core to multicore. Ultimately, MCAPI provides a simple, efficient, and consistent multicore programming model across different types and numbers of cores, operating systems, and physical transports.

Course Objectives
• Introduce multicore standardization issues and why MCAPI benefits system developers [e.g., API for source level portability, small footprint, scalable to one or many cores, allows for more complex functionality on top].
• Learn to leverage open standards and produce programs that take advantage of multicore processors on heterogeneous and homogenous systems.
• Provide in-depth overview of MCAPI fundamentals and operations.
• Explain the functional areas of MCAPI: topology management, modes of communication (connectionless messages, connected channels, packet channels, and scalar channels), non-blocking management, and error handling.
• Analyze MCAPI case studies to better understand the MCAPI programming model.
• Examine programming methods that rely on message passing and shared memory and understand the key API features.
• Discuss strategies that make the most of MCAPI during the migration process, including load balancing, minimizing communication overhead, and shared memory communication versus computation.
• Understand and use MCAPI code templates to further simplify communication between cores on one or more chips when migrating applications.
• Provide hands-on experience with PolyCore Software MCAPI tools to learn how to remap and reconfigure without changing the application’s source code.

Benefits
• Experience a comprehensive training workshop: NCT-450 offers an in-depth overview of fundamental concepts, while offering advanced training and practical advice on developing MCAPI software using C/C++.
• Gain critical insights on how to improve your software’s performance: Students will learn to more easily analyze applications, create topology maps, implement MCAPI communications, and reconfigure applications by using specialized tools to better manage the migration to multicore deployment. Students also will learn to correctly architect, design, and develop efficient parallel applications for multicore processors.
• Enjoy additional hands-on learning: NCT-450 provides laboratory sessions in writing MCAPI programs and exercises on practical parallelization of legacy software. It also includes walk-through laboratory exercises designed to increase your understanding of multicore processors and communications.
• Master PolyCore Software’s MCAPI tools to achieve further time savings for multicore developers: Students will gain an understanding of load balancing, learn to minimize communication overhead to optimize performance, and experience simplified code generation.

Who should attend
System and software architects, developers, team leaders and managers seeking to understand and implement software using MCAPI. Knowledge of the C/C++ programming language and intermediate C/C++ software development experience is a pre-requisite for this course.

About nCore
nCore HPC is a global provider of professional services and systems focused on high performance, low latency and scalability in embedded computing. nCore delivers state-of-the-art solutions to government agencies, high-technology organizations, defense, research, biomedical and financial companies.

nCore is a working group member of the Multicore Association.

Course Registration
Length: 2 Days  Cost: $1895
By E-mail: training@ncorehpc.com

nCore may make changes to specifications & product descriptions at any time, without notice. All trademarks & copyrights are the property of the respective owners. Copyright (C) 2008-2015 nCore HPC LLC.
NCT-500 PGI Accelerator with OpenACC Directives
Two-day intensive training course programming GPUs using PGI Accelerator and OpenACC

Course Overview
This course covers concepts and approaches related to programming GPU processors using OpenACC directives and the PGI Accelerator™ programming model. Extensive coverage of GPU hardware, memories, data transport, and performance optimization enable the student to understand the fundamental aspects of GPU programming. In-depth, hands-on lectures and laboratories demonstrate how to apply OpenACC directives to serial software. Using a directive based approach, students will capitalize on low-cost, high performance GPU computing hardware to improve application performance while reducing maintenance and porting requirements.

Course Objectives
- Correctly identify concurrency opportunities and parallelize algorithms to run on the GPU.
- Install NVIDIA and PGI tools and compile CUDA and OpenACC programs.
- Understand the NVIDIA GPU hardware platform and the underlying technical architecture, including high-throughput SIMD processing and hardware threading architecture concepts.
- Recognize the difference between GPU memory types and the advantages and disadvantages of each.
- Learn to determine the best methods for software development with the OpenACC API.
- Understand the OpenACC command set and its application to C and Fortran codes.
- Learn the specific skills to accelerate applications on x64+GPU platforms with the PGI Accelerator compilers.
- Learn to tune data movement, memory loads and stores, and loop schedules for maximum effect.
- Effectively orchestrate the transport of data to and from GPU memory.
- Learn to meld multicore processors and GPUs to take maximum advantage of modern platform performance.
- Discover how to take advantage of multiple GPUs in the same computer.
- Cover debugging strategies for OpenACC codes.

Benefits
- Offers a detailed overview of fundamental concepts, while providing advanced training and practical advice on GPU programming using OpenACC directives and PGI Accelerator Workstation.
- Teaches everything necessary to start developing high-performance GPU software on Linux, Windows, and Mac platforms using PGI development tools.
- Demonstrates how programmers can gain detailed control over loop mapping, memory allocation, and optimization for the GPU memory hierarchy.
- Shows how to use OpenACC directives to take advantage of massive parallelism, increase throughput, minimize data traffic, and improve program portability.
- Learn to reduce the costs associated with parallel programming by harnessing the potential of GPU processing power.

Who should attend
Software architects, developers, team leaders, and managers seeking to improve their GPU software skills using the OpenACC API and the PGI Accelerator programming model with C or Fortran. Knowledge of computer architectures and intermediate C or Fortran programming, as well as corresponding software development experience are mandatory pre-requisites for this course.

About nCore
nCore HPC is a global provider of professional services and systems focused on high performance, low latency and scalability in embedded computing. nCore delivers state-of-the-art solutions to government agencies, high-technology organizations, defense, research, biomedical and financial companies.

nCore is a working group member of the Multicore Association.

Course Registration
Length: 2 Days  Cost: $1895
By E-mail: training@ncorehpc.com

nCore may make changes to specifications & product descriptions at any time, without notice. All trademarks & copyrights are the property of the respective owners. Copyright (C) 2008-2015 nCore HPC LLC.
## nCore Training Courses 2015 Price List

<table>
<thead>
<tr>
<th>Course</th>
<th>Course Title</th>
<th>Duration</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCT-100</td>
<td>Programming Multicore Processors</td>
<td>3 Days</td>
<td>$2495</td>
</tr>
<tr>
<td>NCT-200</td>
<td>Advanced Multicore Techniques</td>
<td>3 Days</td>
<td>$3495</td>
</tr>
<tr>
<td>NCT-300</td>
<td>Programming GPU Processors</td>
<td>4 Days</td>
<td>$3495</td>
</tr>
<tr>
<td>NCT-450</td>
<td>MCAPI Programming</td>
<td>2 Days</td>
<td>$1895</td>
</tr>
<tr>
<td>NCT-500</td>
<td>PGI Accelerator Programming</td>
<td>2 Days</td>
<td>$1895</td>
</tr>
</tbody>
</table>

### Course Registration Information

- The list price is per student. Minimum enrollment is six students. Inquire about current course discounts.
- Register for a course: info@ncorehpc.com