nCore
NCT-200

nCore

NCT-200 Advanced Multicore Techniques

Three-day intensive training course focused on profiling and optimizting 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 to 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), Open|SpeedShop, 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

THE Multicore ™
ASSOCIATION

intel Software Partner