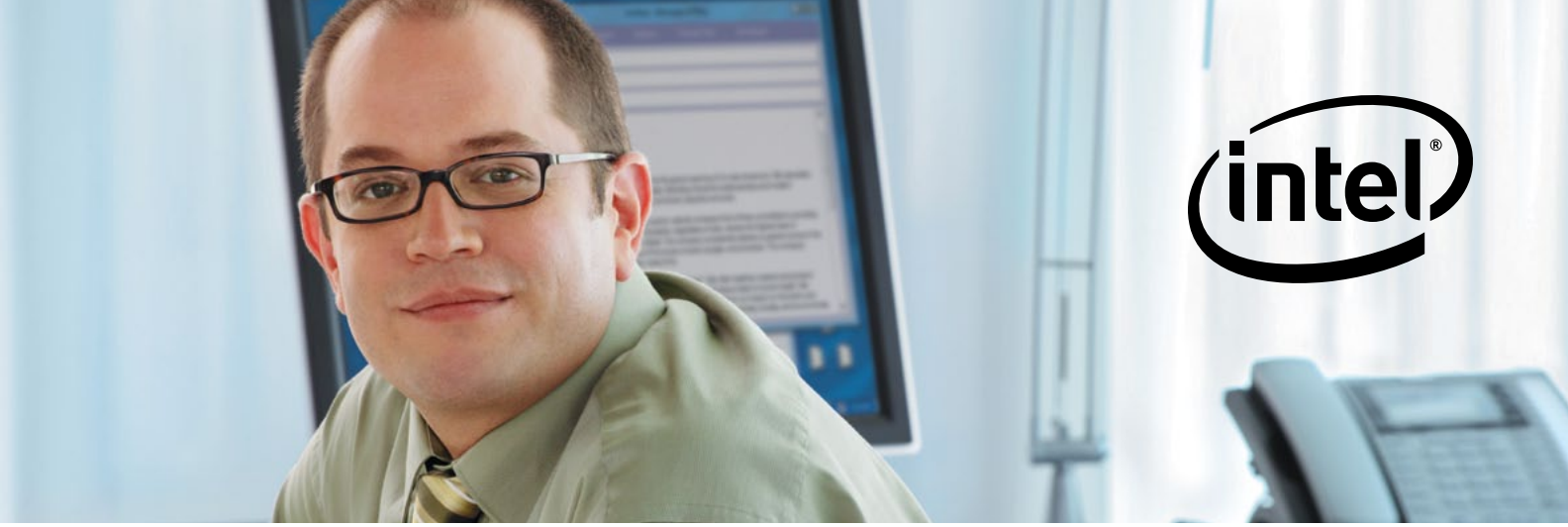


Extending capabilities to better utilize the Windows*-based cluster

RWTH Aachen University* adopts Intel® Parallel Studio to help developers more quickly move forward with multicore applications

Success Brief

Intel® Developer Products

Intel® Parallel Studio for Microsoft Visual Studio*

Research

RWTH AACHEN UNIVERSITY

"Intel® Parallel Studio extends Microsoft Visual Studio* to provide an end-to-end integrated parallelism development environment."

Christian Terboven
Technical Engineer Center for Computing and Communication
RWTH Aachen University

Company	RWTH Aachen University*, one of Germany's foremost technical universities, operates some of the largest high performance computing (HPC) clusters in Europe. Offering both compute and consultative resources to the greater European community of universities and research labs, Aachen's technical staff has become a hub for parallelism. Its Center for Computing and Communication helps ease this transition with consulting and hands-on assistance for clients that include mechanical engineers writing their own code and conducting detailed analysis.
Mission	Aachen's main goal is to move clients to the Windows*-based cluster. With a long history of supporting various high-performance applications, Aachen is also a driving force in the OpenMP community.
Challenge	Move clients from Linux* to Windows* to maximize the benefit of all the cluster processor cores. The university also wants to help clients develop applications based on a proper parallelization strategy, while addressing the many complex issues related to porting and parallelization.
Results	The capabilities of Aachen and its clients were extended, facilitating better use of parallelism.
Impact	Intel® Parallel Studio provided a simple approach to performing analysis and enhanced productivity.

Challenge: why Aachen University benefits from utilizing parallelism

"Everything is parallel today and for the future. Serial programming is dying out," according to Christian Terboven, technical engineer at Aachen's Center for Computing and Communication. With that reality in mind, Aachen typically begins OpenMP* parallelization with runtime analysis to identify opportunities for parallelization of compute-intensive hotspots.

A priority for Aachen's clients is increasing performance on nodes—which increases the likelihood of creating data races and other common parallelization errors. Having the tools to quickly identify data races, evaluate code, and look for performance improvements is critical to helping these advanced developers for whom parallelization is secondary to solving the problems at hand.

Results

Intel Parallel Studio provided better support for debugging, correctness, features, OpenMP 3.0, etc., while integration with Visual Studio 2008* increased clients' parallelism capabilities. These advancements enabled both experts and parallelism novices to move forward with applications developed for multicore, and supported Aachen's educational mission by shortening the parallelism learning curve for clients.

The integration of Intel Parallel Studio into the familiar Microsoft Visual Studio development environment and the ease-of-use of the GUI were pluses, particularly for clients who are competent professional developers, but relatively new to bringing parallelism to their applications..

Intel® Parallel Studio brings comprehensive parallelism to C/C++ Microsoft Visual Studio* application development.

Parallel Studio was created in direct response to the concerns of software industry leaders and developers. From the way the products work together to support the development life cycle to their unique feature sets, Parallel Studio makes parallelism easier and more viable than ever before.

The tools are designed so those new to parallelism can learn as they go, and experienced parallel programmers can work more efficiently and with more confidence. Parallel Studio is interoperable with common parallel programming libraries and API standards, such as Intel® Threading Building Blocks (Intel® TBB) and OpenMP*, and provides an immediate opportunity to realize the benefits of multicore platforms.

Build Applications for Multicore

Intel® Parallel Composer is part of the larger Intel® Parallel Studio and brings an unprecedented breadth of parallelism development options for developers using Microsoft Visual C++*. Its combination of compilers, libraries, and an extension to the Microsoft Visual Studio debugger supports easier, faster multithreading of serial and parallel applications.

Easily Find Memory and Threading Errors

Intel® Parallel Inspector combines threading and memory error checking into one powerful error checking tool. It helps increase the reliability, security, and accuracy of C/C++ applications from within Microsoft Visual Studio*. Intel® Parallel Inspector uses dynamic instrumentation that requires no special test builds or compilers, so it's easier to test code more often.

Optimize Performance and Scalability

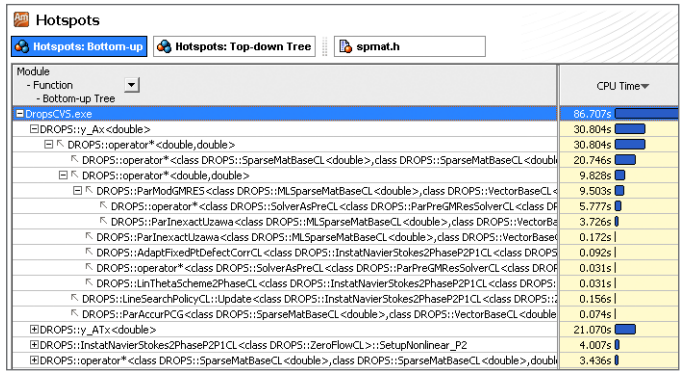
Intel® Parallel Amplifier makes it simple to quickly find multicore performance bottlenecks without needing to know the processor architecture or assembly code. Parallel Amplifier takes away the guesswork and analyzes performance behavior in Windows* applications, providing quick access to scaling information for faster and improved decision making.

How Intel Parallel Studio Assisted

Aachen used Intel Parallel Studio throughout the development life cycle, from runtime analysis and serial tuning, to finding hotspots, debugging, and correctness checking. With Intel Parallel Studio, the Visual Studio environment is extended to make it easier to develop parallel applications, even by those new to parallelism, providing the foundation for applications that can get the most performance on multicore platforms.

Intel® Parallel Composer significantly extended Visual Studio debugger capabilities, providing new debugging views for task parallelism. The tool includes hard-to-find debugger capabilities for OpenMP 3.0. While parallel development can easily cause data races, the problems can be hard to find with traditional debuggers, making Intel Parallel Composer especially important during the build process.

Hotspot-based display of analysis result



Intel® Parallel Inspector allowed comparison of serial and parallel performance profiles. Thread utilization information per function helped users evaluate the scalability and efficiency of their parallelization efforts. Easily accessible data on threading errors was critical for finding common OpenMP programming errors, such as data races.

(OpenMP*-specific) automated data race detection

Relation Sets	ID	Short Description	Severity	Description	Count	
1	1	Read -> Write data-race		Memory write at "main.c":196 conflicts with a prior memory read at "main.c":125 [anti...	1	False
2	2	Write -> Read data-race		Memory read at "jacobi.c":61 conflicts with a prior memory write at "jacobi.c":53 (flow dependence)	70	False
2	3	Write -> Read data-race		Memory read at "jacobi.c":61 conflicts with a prior memory write at "jacobi.c":52 (flow dependence)	70	False
2	4	Write -> Read data-race		Memory read at "jacobi.c":61 conflicts with a prior memory write at "jacobi.c":51 (flow dependence)	70	False
2	5	Write -> Read data-race		Memory read at "jacobi.c":61 conflicts with a prior memory write at "main.c":196 (flow dependence)	70	False
2	6	Write -> Write data-race		Memory write at "jacobi.c":66 conflicts with a prior memory write at "jacobi.c":69 (output dependence)	96	False

Intel® Parallel Amplifier provided a detailed hotspot-based display of analysis results that led users directly to the hotspots impacting application performance. Getting data at the source line level frequently yielded surprising results for users, indicating that this view should be examined before initial parallelization is carried out. .

Optimization Notice

Intel® compilers, associated libraries and associated development tools may include or utilize options that optimize for instruction sets that are available in both Intel® and non-Intel microprocessors (for example SIMD instruction sets), but do not optimize equally for non-Intel microprocessors. In addition, certain compiler options for Intel compilers, including some that are not specific to Intel micro-architecture, are reserved for Intel microprocessors. For a detailed description of Intel compiler options, including the instruction sets and specific microprocessors they implicate, please refer to the "Intel® Compiler User and Reference Guides" under "Compiler Options." Many library routines that are part of Intel® compiler products are more highly optimized for Intel microprocessors than for other microprocessors. While the compilers and libraries in Intel® compiler products offer optimizations for both Intel and Intel-compatible microprocessors, depending on the options you select, your code and other factors, you likely will get extra performance on Intel microprocessors.

Intel® compilers, associated libraries and associated development tools may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include Intel® Streaming SIMD Extensions 2 (Intel® SSE2), Intel® Streaming SIMD Extensions 3 (Intel® SSE3), and Supplemental Streaming SIMD Extensions 3 (Intel® SSSE3) instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors.

While Intel believes our compilers and libraries are excellent choices to assist in obtaining the best performance on Intel® and non-Intel microprocessors, Intel recommends that you evaluate other compilers and libraries to determine which best meet your requirements. We hope to win your business by striving to offer the best performance of any compiler or library; please let us know if you find we do not.

Notice revision #20101101

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing.

For more information on performance tests and on the performance of Intel products, visit http://www.intel.com/performance/resources/limits.htm.

1210/BLA/CMD/PDF 322256-001US