

INNOVATIVE THREADING ASSISTANT

Intel® Parallel Advisor 2011

Product Brief

Intel® Parallel Advisor 2011



“The Intel® Parallel Advisor design approach was instrumental in introducing parallelism into our code. The Survey feature helped improve our code by finding areas in our serial code that took a lot of CPU time, and where our code would benefit from parallelism.”

*William H. Orttung
Emeritus Professor of Chemistry
University of California, Riverside*

Simplify, Demystify, and Speed Threading Design

Intel® Parallel Advisor, a threading assistant tool for Microsoft Visual Studio* C++ developers, helps identify where to add parallelism to existing source code. It guides developers and architects through the process of exploring parallelism, identifying those areas with the greatest parallelism potential. Intel Parallel Advisor also provides tools for proposing parallelization, and for evaluating the performance and correctness of the proposed parallelization. This insight helps developers make better design decisions, providing information on the consequences of those choices and suggesting ways to resolve issues at the design stage—before major effort has been committed.

Streamline parallelization of your applications

- Get step-by-step guidance for modeling parallelism in your applications
- Make better design decisions by analyzing parallelization proposals
- Model applying parallel code to applications at points of greatest impact
- Create, construct, devise, and develop a parallel plan
- Develop standards-based, cross-platform solutions designed to scale for multicore and manycore
- Save time, increase productivity, and speed time to market

Empowers architects with parallelism design insight and analysis for best results

With the Intel Parallel Advisor workflow approach, architects get guidance to, and tools, for implementing parallelism and exploiting the power of multicore.

Discover, analyze, check, and add threading

- Identifies where application architectures can benefit most from parallelism
- Uses dynamic profiling to identify key outer loops and recursions that will benefit most from parallelism
- Uses pre-threading performance model guides programmers to address overheads at the design stage to avoid costly rework
- Guides resolution of key data conflicts and assist programmers in introducing appropriate synchronization
- Identifies changes required including locks, potential race conditions, and hotspots, to add tasks (threads)
- Summarizes the places that need to change, and the impact of those changes, to guide the transition to parallel code

Intel Parallel Advisor analyzes the executing serial program as developers work through the methodology



Figure 1: Design Workflow

Survey Target—Focuses effort where it will pay off and targets coarse-grain parallelism

Annotate Sources—Inserts annotations to sources to describe parallel experiments

Check Suitability—Evaluates the performance impact and implication of the parallel experiments

Check Correctness—Finds and resolves conflicts before parallelism is applied. Reduces debugging problems by defining safe and solid paths to parallelism.

Survey Target—Focuses on the hot call trees and loops as locations to experiment with parallelism. Developers can drill into the source code.

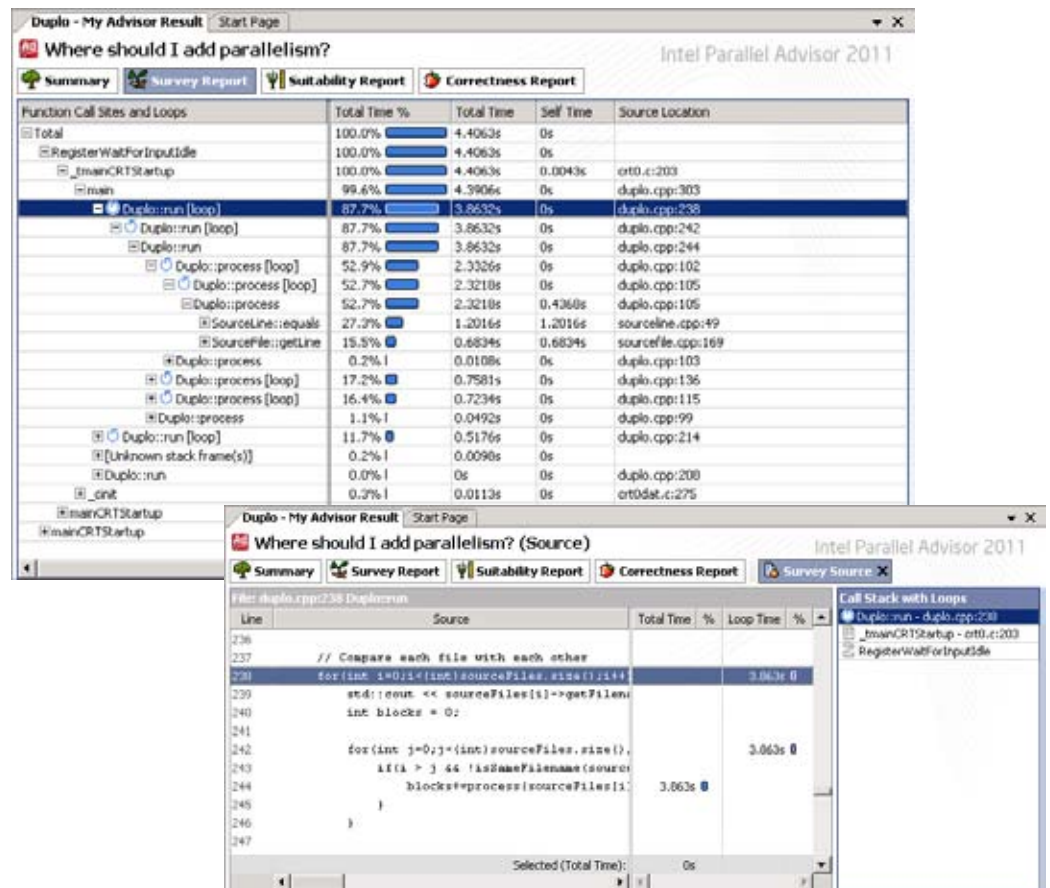


Figure 2: Survey

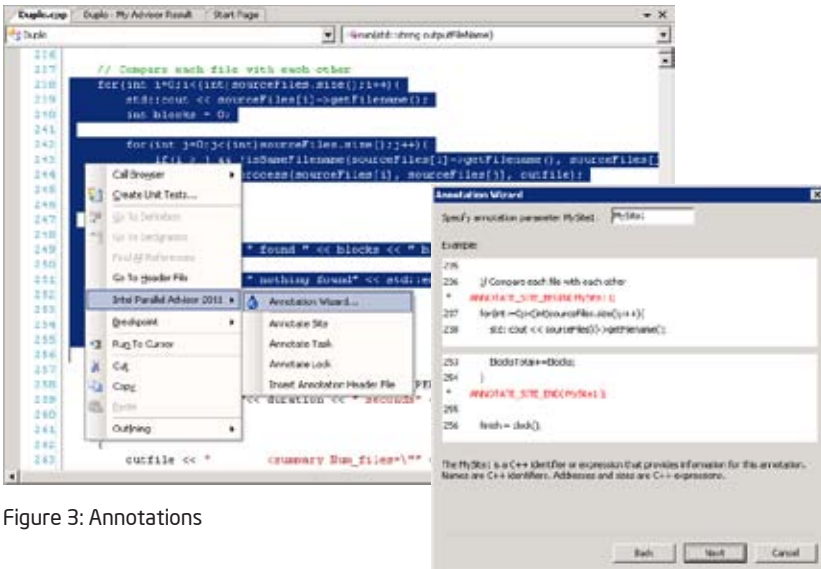


Figure 3: Annotations

Annotations—Inserts Intel Parallel Advisor annotations into your sources to describe the parallel experiments. For example, you can experiment with parallelism by modeling the hot calls and loops identified during survey analysis with Intel Parallel Advisor annotations.

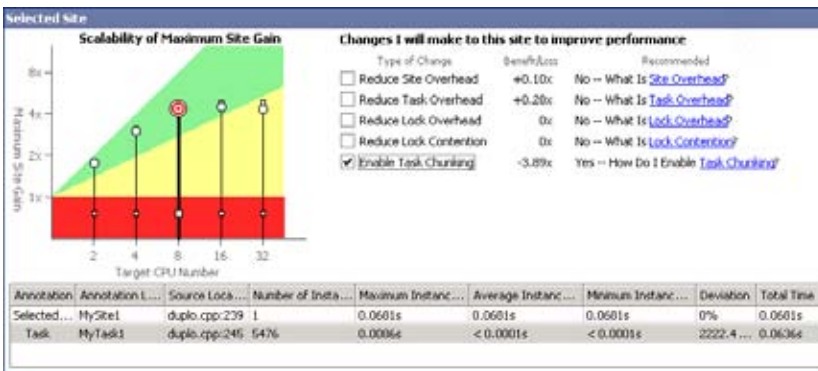
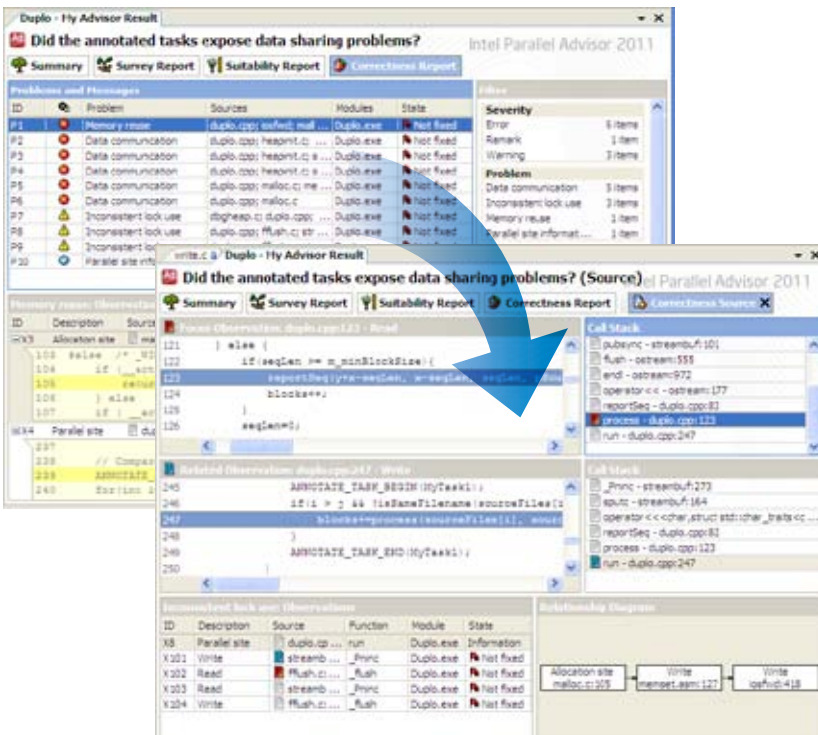


Figure 4: Check Suitability

Suitability—Evaluates the performance of your parallel experiment by displaying the performance projection for each parallel site and how each site's performance impacts the entire program.



Correctness—Identifies the data races or deadlocks found in the experiment. Evaluates the data-sharing problems that prevent correct parallel execution.

Figure 5: Correctness

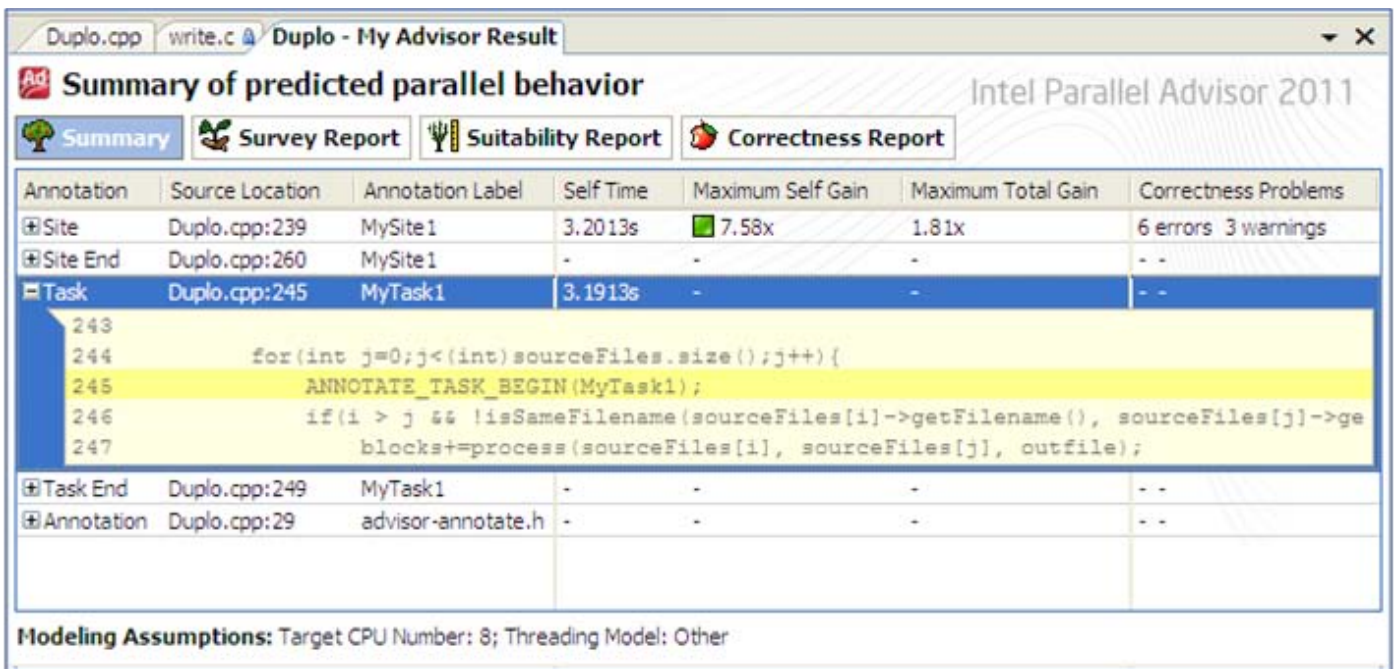


Figure 6: Summary Report

Summary Report—Quickly identifies the parallel experiments to focus on projected performance gains and the number of correctness problems.

- Summarizes and compare the parallel experiments and identify where performance gains are easiest
- Lists source location of annotations to guide conversion to final parallel code

Create robust, feature-rich threaded applications faster and more cost-effectively with Intel Parallel Advisor

- Use with any native C/C++ application
- Interoperable with Microsoft* and Intel® Compilers
- Ideal for applications that are not easily parallelized

System Requirements

Microsoft Visual Studio 2005*, 2008*, 2010* (except the Express Edition)

For the latest system requirements, go to: www.intel.com/software/products/systemrequirements/.

Support

Purchase of Intel® Parallel Studio products include Premium Support service which allows you to submit questions, access to product updates, and technical documentation.

For more information, go to: <http://software.intel.com/en-us/articles/intel-parallel-studio/>.

"I would recommend Intel® Parallel Advisor to any Microsoft Visual Studio* C++ programmer interested in parallel programming. Intel Parallel Advisor's methodology and steps helped simplify and speed the planning and production phases."

*Brian Reynolds
Brian Reynolds Research*

Download a Trial Version Today

Evaluation copy available at:
www.intel.com/software/products/ParallelStudio/

The Ultimate All-in-One Performance Toolkit—Intel® Parallel Studio 2011

Designed for today's serial applications and tomorrow's software innovators

Intel brings simplified threading to Microsoft Visual Studio* C++ developers with a complete productivity solution designed to optimize serial and new threaded applications for multicore and scale for manycore.

INNOVATIVE THREADING ASSISTANT

Intel® Parallel Advisor 2011: Demystify and speed threaded application design.

COMPILER AND THREADED LIBRARIES

Intel® Parallel Composer 2011: Develop effective applications with a C/C++ compiler and advanced threaded libraries.

MEMORY AND THREADING ERROR CHECKER

Intel® Parallel Inspector 2011: Ensure application reliability with proactive parallel memory and threading error checking.

THREADING AND PERFORMANCE PROFILER

Intel® Parallel Amplifier 2011: Quickly find bottlenecks and tune threaded applications for scalable multicore performance.

