



THREADING ASSISTANT



Intel® Advisor XE 2013

Product Brief

Top Features

- Profiling identifies where an application will benefit most from parallelism
- Compare the effort and benefit of alternatives before you invest in implementation
- Guides resolution of key data conflicts. Assists in adding synchronization.

Intel® Advisor XE is included in all Intel® Studio XE products:

- Intel® Parallel Studio XE
- Intel® C++ Studio XE
- Intel® Fortran Studio XE
- Intel® Cluster Studio XE

OS Support:

- Windows*
- Linux*

“Intel® Advisor XE can be invaluable in developing the understanding required to parallelize existing code. It assists with identifying opportunities, designing tests, modeling scenarios and revealing flaws.”

Matt Osterberg,
Senior Software Engineer,
Vickery Research Alliance

Simplify, Demystify and Speed Threading Design

Intel® Advisor XE is a threading assistant for C, C++, C# and Fortran. It guides developers through threading design, automating analyses required for fast and correct implementation.

Explore Alternatives Before Implementation

Intel Advisor XE runs a performance analysis to identify opportunities for loop, task and recursive parallelism. Next, use simple code annotations to quickly describe potential parallelism and let Intel Advisor XE estimate the performance and suggest changes. Then run a correctness report to discover critical synchronization issues. Looping through this design process lets you quickly explore alternatives and make tradeoffs. Intel Advisor XE helps you make better design decisions by accurately projecting performance and identifying synchronization errors. You get faster parallel performance while avoiding costly design errors.

Parallelize Without Adding Schedule Risk

Annotations describe your design to Intel Advisor XE letting it give you performance and correctness information, but make no changes to your compiled code. You can continue to release product updates during the design of your parallelism. All your test cases continue to work. Verify that your application is stable and correct before you go to the final step - implementing parallelism.

Implement With Productive Parallel Programming Models

All the Intel® Studio XE products include a selection of productive parallel models such as Intel® Threading Building Blocks (Intel® TBB) and Intel® Cilk™ Plus. Implementing parallelism at a higher level yields scalable and reliable parallelism with fewer lines of code. Task-based algorithms, containers and synchronization primitives simplify parallel application development. The task scheduler for Intel® TBB and Intel® Cilk™ Plus dynamically maps tasks to threads to load balance, preserve cache locality and increase performance. This lets you develop faster and deliver more performance with less code to maintain.

What are the performance implications of the annotated sites? Intel Advisor XE 2013

Summary | Survey Report | Annotation Report | Suitability Report | Correctness Report

All Sites

Maximum Program Gain For All Sites: 5.31x

Target CPU Number: 8 | Threading Model: Intel TBB

Annotation Label	Source Location	Maximum Site Gain	Maximum Total Gain	Average Instance Time	Total Time
mainLoop	primes_ann...	5.96x	5.31x	4.6862s	4.6862s

Selected Site

Scalability of Maximum Site Gain

Changes I will make to this site to improve performance

Type of Change	Benefit if Checked	Loss if Unchecked	Recommended
<input type="checkbox"/> Reduce Site Overhead	0x		No
<input type="checkbox"/> Reduce Task Overhead	0x		No
<input type="checkbox"/> Reduce Lock Overhead	0x		No
<input type="checkbox"/> Reduce Lock Contention	0x		No
<input checked="" type="checkbox"/> Enable Task Chunking		5.80x	Yes

1) Select the changes you will make
2) See the projected scalability

Intel Advisor XE's suitability analysis gives you a performance estimate before you invest significant effort in implementation. Implement only the options that have a high return on investment.

Top Features

1. Survey Target
Where should I consider adding parallelism? Locate the loops and functions where your program spends its time, and functions that call them.
Collect Survey Data
View Survey Result

2. Annotate Sources
Add Intel Advisor XE annotations to identify possible parallel tasks and their enclosing parallel sites.
Steps to annotate
View Annotations

3. Check Suitability
Analyze the annotated program to check its predicted parallel performance.
Collect Suitability Data
View Suitability Result

4. Check Correctness
Predict parallel data sharing problems for the annotated tasks. Fix the reported sharing problems.
Collect Correctness Data
View Correctness Result

5. Add Parallel Framework
Steps to replace annotations
View Summary

Workflow For Effective Design

Intel Advisor XE's workflow panel guides you through the steps to successfully add effective threading to your application. Implementation is delayed until step 5. This lets you use the active code base and continue to release product updates during the design phase (steps 1-4).

Where should I add parallelism? Intel Advisor XE 2013

Function Call Sites and Loops	Total Time %	Total Time	Self Time	Source Location
_tmainCRTStartup	100.0%	2.0903s	0s	crtexe.c:552
wmain [loop]	100.0%	2.0903s	0s	
wmain	90.7%	1.8950s	0s	
IsPrime [loop]	90.2%	1.8850s	0s	

Loop Tasks Annotations Example (hide)

```
// Place before the loop control statement for the first ta
ANNOTATE_SITE_BEGIN( <site_name> );
// loop control statement
// Use the task-begin and task-end annotations for site.
// If your site contains a single loop task or function
ANNOTATE_TASK_BEGIN( <task_name> );
// loop body
ANNOTATE_TASK_END();
```

What Should I Do Next?
Topic: Loop, One or More Tasks (bound...
Copy

1. Survey - Search for Parallel Sites

Start by measuring your app to see where it will benefit from parallelism. There is no point in expending effort unless it will pay off with better performance.

2. Annotate Your Source

Insert annotations in your source code to quickly sketch out potential parallelism. Intel Advisor XE provides coding examples and lets you easily cut and paste them into your source.

What are the performance implications of the annotated sites? Intel Advisor XE 2013

Maximum Program Gain For All Sites: 5.31x

Annotation Label	Source Location	Maximum Site Gain	Maximum Total Gain	Average Instance Time	Total Time
mainLoop		5.31x	5.31x	4.6862s	4.6862s

Scalability of Maximum Site Gain

Changes I will make to this site to improve performance

Type of Change	Benefit if Checked	Loss if Unchecked	Recommended
Reduce Site Overhead	0x	0x	No
Reduce Task Overhead	0x	0x	No
Reduce Lock Overhead	0x	0x	No
Reduce Lock Contention	0x	0x	No
Enable Task Chunking	5.80x	5.80x	Yes

1) Select the changes you will make
2) See the projected scalability

3. Check Performance - Compare Alternative Designs

Check the performance and scalability of the design you sketched out. Is it as fast as you thought? How will it scale on systems with more processors? Intel Advisor XE recommends possible changes and estimates their performance benefit.

Did the annotated tasks expose data sharing problems? Intel Advisor XE 2013

ID	Problem	Site Name	Sources	Modules	State
P1	Parallel site information	solve	nqueens_annotated.cpp	2_nqueens_annotate...	New
P2	Data communication	solve	nqueens_annotated.cpp	2_nqueens_annotate...	Confirmed
P3	Memory reuse	solve	nqueens_annotated.cpp	2_nqueens_annotate...	New

Data communication: Code Locations

ID	Description	Source	Function	Module	State
X2	Parallel site	nqueens_annotated.cpp:113	solve	2_nqueens_annotated.exe	Confirmed
X3	Read	nqueens_annotated.cpp:90	setQueen	2_nqueens_annotated.exe	Confirmed

```
111 int * queens = new int[size]; //array representing queens placed on a ch
112
113 ANNOTATE_SITE_BEGIN(solve);
114 for(int i=0; i<size; i++) {
115     // try all positions in first row
88 //ANNOTATE_LOCK_ACQUIRE(0);
89 //ADVISOR COMMENT: This is a race condition because multiple tasks ma
90 mrofSolutions++; //Placed final queen, found a solution!
91 //ANNOTATE_LOCK_RELEASE(0);
92 }
```

4. Check Correctness

Do you have any data sharing problems that can lead to deadlocks or races? Find and fix them now before you implement the parallelism.

Intel Advisor XE gives you a list of errors and shows you a snippet of the code at all the related code locations, with click-through navigation to the actual source location.

Parallel Programming Models

C, C++	- Intel® TBB - Intel® Cilk™ Plus - OpenMP	C#	- Microsoft TPL*
		Fortran	- OpenMP

5. Implement - Choose A Parallel Programming Model

Implement parallelism using a productive parallel programming model that yields scalable and reliable parallelism with fewer lines of code.

Intel® Threading Building Blocks (Intel® TBB) has a rich set of abstractions. Intel Cilk Plus is simple. OpenMP is compatible with legacy code. Microsoft TPL* is designed for C#.

Details

Speed For Your App

The addition of threading can dramatically improve your application's performance with today's multicore processors. Using Intel Advisor XE, you can easily explore multiple threading designs and pick the one that gives the best return on your time investment.

Simplify

Ideally Intel Advisor XE would automatically add threading to your application and make it run faster. Unfortunately, the technology for doing this well is beyond the current *state of the art*. Instead, Intel Advisor XE is designed for a software architect or developer. It automates the analyses required and gives the developer the information needed to productively add effective threading to an existing application.

Demystify

Intel Advisor XE's workflow window sequences and tracks the steps required for effective threading design. Whether you are new to threading or an experienced professional, this checklist approach keeps you organized. Once the design is final, Intel Advisor XE provides cut and paste templates to implement threading using a one of several parallel programming models.

Speed Implementation - Design Threading Without Impacting Normal Development

The delayed implementation workflow means that the threading design, performance projection and error analysis can proceed in parallel with normal release development. During the design phase no changes are made to the source that impact code generation. You can continue to build, test and release your application while you design and test the threading. Implement threading only when you are confident that your app is stable, the threading design is solid and the timing is right for your release schedule. Intel Advisor XE's automation makes developers and architects more productive, reducing the time needed to add effective threading to an application.

Open Parallel Programming Models

Intel Advisor XE gives you a choice of open and industry standard parallel programming models.

- Intel® TBB - commercial and open source
- Intel® Cilk™ Plus - Intel® and GCC* compilers (currently in the [development branch for GCC](#) watch for it the release.)
- Intel® OpenMP - Industry standard
- Microsoft TPL* - Microsoft* standard for C#

What's New?

Feature	Benefit
Linux* OS	Intel Advisor XE is now available for both Windows* and Linux*.
More languages: C, C++, C# and Fortran	New support for C# and Fortran enables developers to use Intel Advisor XE on more applications.
More parallel models: Intel® TBB, Intel® Cilk Plus, TPL* and OpenMP*	Choose a parallel programming model that best fits your needs.
Command line, standalone user interface, Visual Studio* Integration	Work the way that is most natural and productive for you. Use Intel Advisor XE integrated with Microsoft Visual Studio* or with a standalone interactive user interface or from a command line.

Purchase Options: Language Specific Suites

Intel Advisor XE is not sold separately. It is only available as a part of a product suite. Single or multi-user licenses along with volume, academic, and student discounts are available.

Suites >>		Intel® Cluster Studio XE	Intel® Parallel Studio XE	Intel® C++ Studio XE	Intel® Fortran Studio XE	Intel® Composer XE	Intel® C++ Composer XE	Intel® Fortran Composer XE
Components	Intel® C / C++ Compiler	●	●	●		●	●	
	Intel® Fortran Compiler	●	●		●	●		●
	Intel® Integrated Performance Primitives ³	●	●	●		●	●	
	Intel® Math Kernel Library ³	●	●	●	●	●	●	●
	Intel® Cilk™ Plus	●	●	●		●	●	
	Intel® Threading Building Blocks	●	●	●		●	●	
	Intel® Inspector XE	●	●	●	●			
	Intel® VTune™ Amplifier XE	●	●	●	●			
	Intel® Advisor XE	●	●	●	●			
	Static Analysis	●	●	●	●			
	Intel® MPI Library	●						
	Intel® Trace Analyzer & Collector	●						
	Rogue Wave IMSL* Library ²							●
Operating System ¹	W, L	W, L	W, L	W, L	W, L	W, L, O	W, L, O	

Note: ¹ Operating System: W=Windows, L= Linux, M= OS X*. ² Available in Intel® Visual Fortran Composer XE for Windows with IMSL*

³ Not available individually on OS X, it is included in Intel® C++ & Fortran Composer XE suites for OS X

Technical Specifications

Specs at a Glance	
Processor Support	Intel® processors and compatible processors when analyzing applications containing Intel® instructions.
Operating Systems	Windows* and Linux*
Development Tools and Environments	Compatible with compilers from vendors that follow platform standards (e.g., Microsoft*, GCC, Intel compilers). Can be integrated with Microsoft Visual Studio* 2008, 2010 and 2012. See http://www.intel.com/software/products/systemrequirements for the latest details.
Programming Languages	C, C++, C#, Fortran.
System Requirements	See http://www.intel.com/software/products/systemrequirements for details
Support	All product updates, Intel® Premier Support services and Intel® Support Forums are included for one year. Intel Premier Support gives you secure, web-based, engineer-to-engineer support.
Community	Join the Intel® Support Forums community to learn, contribute, or just browse! http://software.intel.com/en-us/forums

	<p>Learn more about Intel Advisor XE: Click or enter the link below: http://intel.ly/intel-advisor-xe</p> <ul style="list-style-type: none"> Or scan the QR code on the left 		<p>Download a free 30-day evaluation</p> <ul style="list-style-type: none"> Click or enter the link below: http://intel.ly/sw-tools-eval Click on 'Compilers and Libraries' link
---	--	--	--

Optimization Notice

Notice revision #20110804

Intel's compilers 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 SSE2, SSE3, and 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. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.