

Intel® Parallel Amplifier

Product Brief

Intel® Parallel Amplifier



"I was delighted when it pointed me to the right source line that was taking much of the time. I made the change, and voila, our app is now almost 10 times faster."

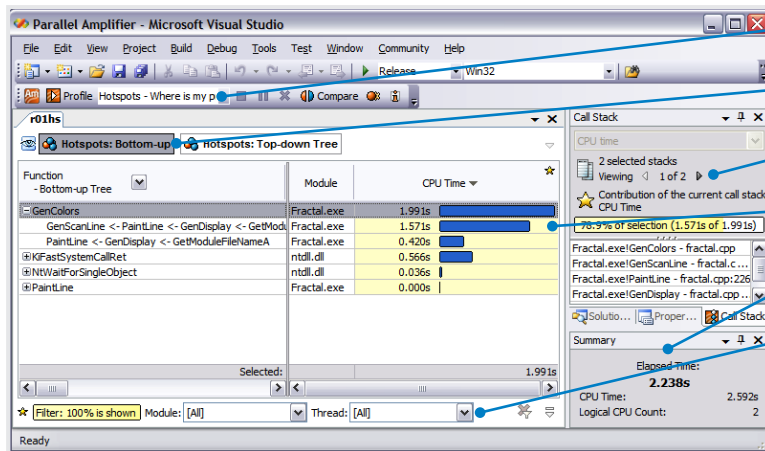
Anonymous beta tester describing Intel® Parallel Amplifier

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. Intel 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.

Fine-tune for optimal performance, ensuring cores are fully exploited and new capabilities are supported.

- Make significant performance gains that impact customer satisfaction
- Increase application headroom for richer feature sets and next-gen innovation
- Find performance problems quickly and easily
- Scale applications for multicore



Choose analysis type

View call stack bottom-up or top-down

Call stacks for selected function

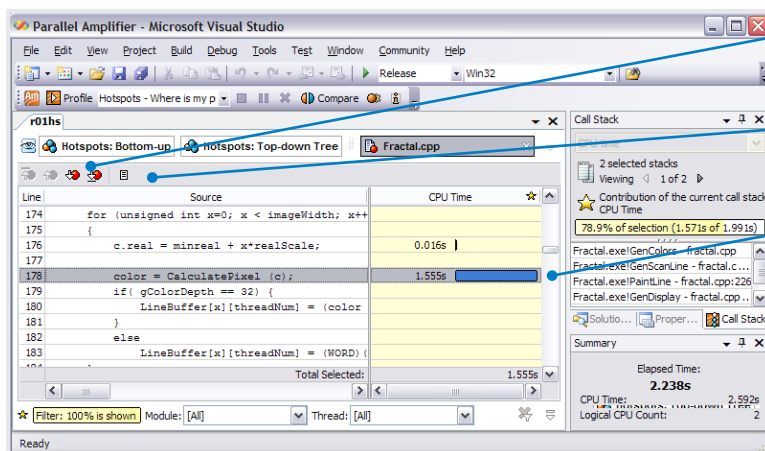
CPU times for hot functions

Total elapsed and CPU time

Filter results

Hotspot Analysis: Where is my app spending time?

Find the functions in your application that consume most of the time. This is where you'll want to tune or add parallelism to make your program faster. Intel Parallel Amplifier also shows the stack, so you know how the function is being called. For functions with multiple calling sequences, this lets you see if one of the call stacks is hotter than the others.



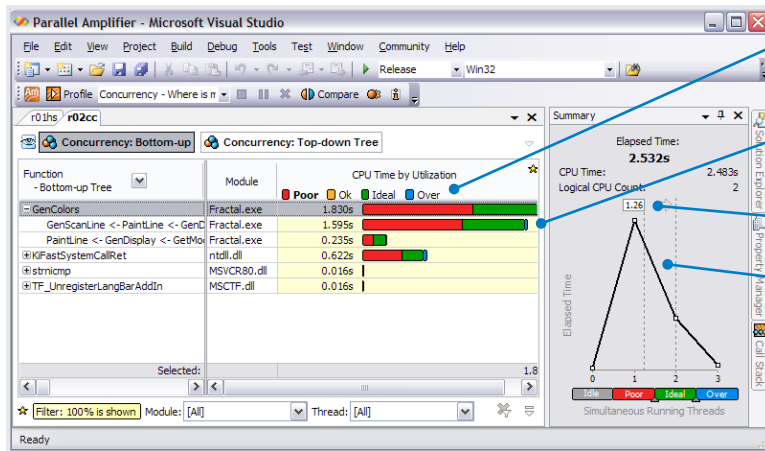
Click to go to the next hottest spot in this function

Click to open the file editor at this location

When you double click a function in any analysis, it opens the source to the hottest spot

Source View: See the results on your source

Source view shows you the exact location on your source. Just double-click on the function names in any of the analysis views to see the source.



Tune core utilization

Green = Ideal Red = Poor

Length of bar is time, color is utilization while the function is running

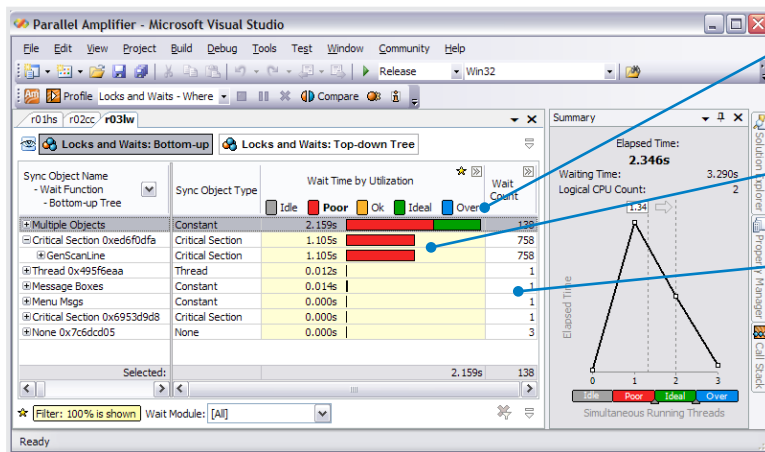
Average CPU utilization

Most of the time this app only uses one core

Call stack information (not shown) is available

Concurrency Analysis: When are cores idle?

Like hotspot analysis, concurrency analysis finds the functions where you are spending the most time. But it also shows you how well you are utilizing multiple cores. Color indicates the core utilization while the function is running. A green bar means all the cores are working. A red bar means cores are underutilized. When there is red, add parallelism and get all the cores working for you. This helps you ensure application performance scales as more cores are added.



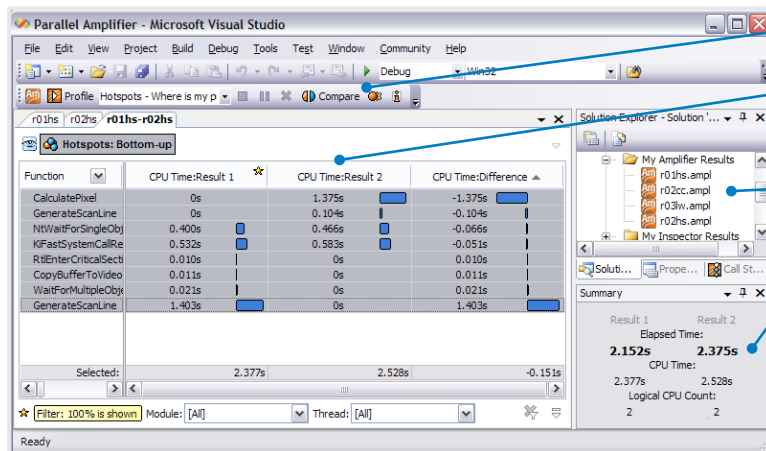
Length of bar is wait time, color is number of cores utilized during the wait

Waiting with underutilized cores hurts performance

Wait count helps identify interesting waits

Locks and Waits Analysis: Where are the bad waits?

Waiting too long on a lock is a common source of performance problems. It's not bad to wait while all the cores are busy (green). It is bad to wait when there are unused cores available (red).



Compare two results

See the times and the difference for each function

Use any of your previous results

Summary of the change

Compare Results: Quickly see what changed

This gives you a fast way to check progress when tuning and also makes a handy regression analysis.

Features

- Fully integrated with Microsoft Visual Studio*
- Supports Microsoft* and Intel® Compilers
- Works with all models for parallelism offered by Intel® Parallel Studio and Microsoft Visual Studio*, Intel® Threading Building Blocks (Intel® TBB), OpenMP*, and WinAPI.
- Intuitive performance profiler specifically designed for threaded applications
- Find application hotspots and view them on the source
- Tune parallel applications for scalable performance using concurrency analysis
- Use locks and waits analysis to find critical waits that limit parallel performance
- Compare results to quickly see what changed or find regressions
- Build applications that automatically scale for manycore

System Requirements

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

Support

Intel Parallel Studio products include access to community forums and a knowledge base for all your technical support needs, including technical notes, application notes, documentation, and all product updates.

For more information, go to
<http://software.intel.com/sites/support/>

Beta Versions Available Now

Download and register for the user forums at:
www.intel.com/software/ParallelStudioBeta/

**"Intel® Parallel Amplifier helped us improve our
CPU cores throughput almost 20%."**

*Anonymous beta tester describing
Intel® Parallel Amplifier*

Intel® Parallel Studio

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

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

Intel® Parallel Studio: Create optimized serial and parallel applications with the ultimate all-in-one parallelism toolkit

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

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

Intel® Parallel Amplifier: Quickly find bottlenecks and tune parallel applications for scalable multicore performance

