

# **Introduction to the Intel<sup>®</sup> VTune<sup>™</sup> Performance Analyzer 6.0**

This white paper was written for Intel Corp. The document was “pre-sales” collateral. As such, its job was to tell the reader what was wonderful about the product and entice the reader to learn more about it. Therefore, the paper did not contain a lot of technical detail or instructions about how to use the product.

# **Introduction to the Intel<sup>®</sup> VTune<sup>™</sup> Performance Analyzer 6.0**

**Find hotspots, identify  
performance issues, and  
tune application performance**

Revision 1.0  
December 2001

## **Introduction to the Intel® VTune™ Performance Analyzer**

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Intel disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein, except that a license is hereby granted to copy and reproduce this document for internal use only.

Intel, the Intel logo, VTune, Pentium, Xeon, and Celeron are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 2002, Intel Corporation. All rights reserved.

# Contents

<b>Introduction .....</b>	<b>4</b>
<b>Identifying Performance Issues.....</b>	<b>4</b>
Finding Hotspots.....	5
Analyzing Function Caller/Callee Relationships.....	7
Monitoring Performance Counter Activity.....	8
Suggesting Code Modifications.....	9
<b>Obtaining the Intel® VTune™ Performance Analyzer.....</b>	<b>10</b>

## Introduction

Clock speeds get faster. Improved microarchitectures offer larger caches and more powerful machine-level instructions. Multiple-processor systems increase available computing resources. However, all these enhancements may be wasted if software routines make inefficient use of these valuable computing resources. Execution efficiency is especially important for performance-critical applications, where speed is paramount.

Software developers strive to write efficient, elegant code. However, even the best computer programmer cannot always pinpoint the cause of performance problems, and may waste hours or days on misdiagnoses. The Intel® VTune™ Performance Analyzer can help.

The VTune Performance Analyzer<sup>1</sup> assists in finding performance bottlenecks by:

- Identifying “hotspots” or bottlenecks within the application
- Analyzing the caller/callee relationships of functions
- Monitoring runtime performance by using hardware and software counters
- Identifying opportunities where modifying the code can help generate faster executables.

The intuitive graphical interface used by the VTune Performance Analyzer lets programmers easily analyze code performance. They can use their knowledge of the application’s algorithms to interpret the VTune Performance Analyzer results and optimize the code, leading to improved software application performance.

## Identifying Performance Issues

Software performance problems have a number of causes. You must use the right data collection technique or use a combination of collectors, based on their characteristics, to collect the type of data that will help address the problem you are trying to isolate. For example, to see if server transactions are adding to the overhead, you may want to use the counter monitor data collector. However, if you need to determine which functions are causing the most overhead, you might want to use the call graph feature of the VTune Performance Analyzer.

Using a combination of the various VTune Performance Analyzer data collection features, you can:

- Use sampling to find hotspots or bottlenecks in your program’s activity.
- Use a call graph to identify the most time-consuming function calls and call sequences within your code. The call graph also allows you to determine how the various modules running in a complete application affect the performance of each other.
- Determine how your system resources, such as processor time, are being used and determine how your application is executing with respect to processor events (such as cache misses).

The Tuning Assistant is always available to offer insightful suggestions about why your code might be inefficient and how you can improve it.

---

<sup>1</sup> The Intel VTune Performance Analyzer assumes the software being analyzed is running on an Intel processor, which includes but is not limited to, Pentium®, Pentium Pro, Pentium II, Pentium II Xeon™, Celeron®, Pentium III, Pentium III Xeon, Mobile Intel Pentium III Processor –M, Pentium 4, and Intel Xeon. Also, the VTune Performance Analyzer requires Microsoft Internet Explorer\* version 5.5 or later.

The VTune Performance Analyzer saves the data it collects and the data collection configuration information. This makes it easy for you to compare a baseline run against optimized code to see the performance impact of your code modifications.

### ***Finding Hotspots***

A “hotspot” or bottleneck is a section of code in which there is significant activity, where “activity” is defined as execution time or processor events that affect performance, such as branch mispredictions, cache misses, or various types of SIMD instructions being retired. Hotspots can be considered optimization opportunities.

When analyzing application performance, you should concentrate on hotspots. However, a hotspot does not necessarily indicate a problem; it may simply indicate where the application spends a majority of its time. Finding the hotspots in your program can be the first step in identifying an important performance issue or bottleneck. For example:

- Sometimes the hotspot itself can be optimized to yield a significant performance improvement.
- Other times, looking at a hotspot in detail will suggest that a change in algorithm or data structure might improve the performance of your program.
- Finding a hotspot where you do not expect one may expose a bug in your program logic that affects the application performance.

The VTune Performance Analyzer finds hotspots by using statistical sampling of the execution address. This approach has several benefits:

- No code changes or instrumentation are required.<sup>2</sup>
- The profiling is system wide, including ring 0.
- The associated low overhead and minimal intrusion (it does not modify binary files) results in a high level of validity.

By analyzing the sampling results, you can determine how to modify your source code to use processor time and processor-specific features more efficiently. If your application is multi-threaded, the VTune Performance Analyzer reports sample breakdowns by thread.

Two types of sampling are possible:

- Time-based sampling (TBS) – collects samples of active instruction addresses at regular intervals using an OS timer.
- Event-based sampling (EBS) – interrupts the processor after a specified number of processor events and stores the program counter. For example, you can define hotspots using events such as mispredicted branches, cache misses, or instructions retired.

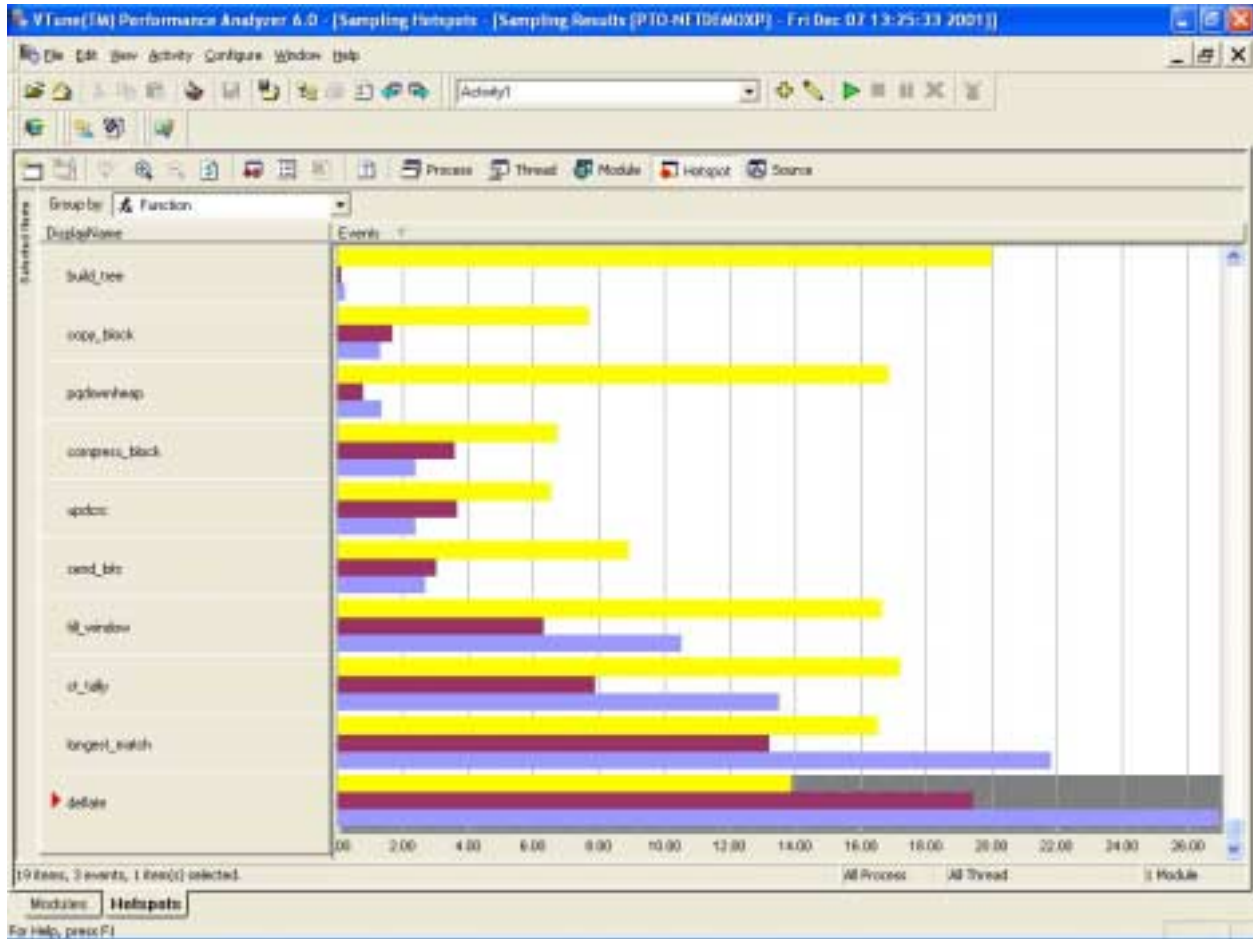
In addition to this flexibility, you can also export your data collection results to Excel\*, for reporting ease.

Figure 1 shows some sampling output.

---

<sup>2</sup> A “standard” release build may not be sufficient, as you do need access to a build that includes symbols.

## Introduction to the Intel® VTune™ Performance Analyzer



**Figure 1. Sampling results from the Quick Performance Analysis (QPA) Wizard help pinpoint hotspots in your code**

In Figure 1, the maroon and blue bars representing the `deflate` function are fairly long compared to all the other functions. These long bars indicate that this function had a relatively high number of instructions retired and took a relatively long time to execute. Figure 1 is drilled down to the “Hotspot” view of sampling. You can also view sampling at the process, thread, and module levels.

Figure 2 shows how you can use the VTune Performance Analyzer to drill down from process to thread to module to source code views, narrowing your focus to the exact lines of code that form the hotspot.

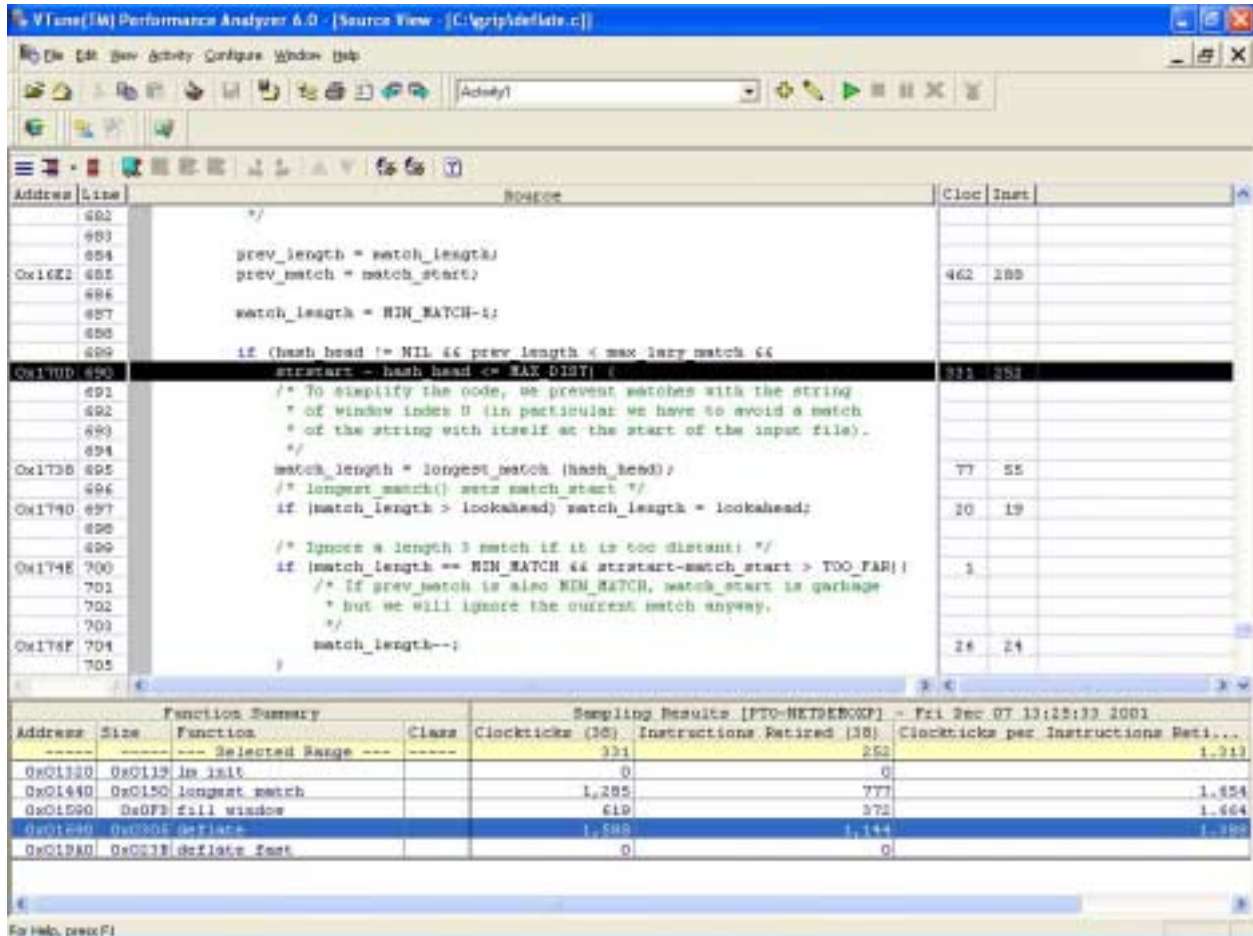


Figure 2. Drilling down to a hotspot in the source code enables you to see exactly which lines of code are slowing down performance

When viewing code, you can view source code, the disassembly of your executable, or both.

## Analyzing Function Caller/Callee Relationships

The call graph information includes the following:

- Number of calls for each caller and callee
- Time spent in each function or method
- Time each function spends blocked or waiting
- Critical paths through the calling hierarchy
- Functions representing the top  $n\%$  time spent, where  $n$  is user-specified.

Historically, gaining this type of information required time-consuming code changes. In contrast, the call graph collector of the VTune Performance Analyzer generates a graphical representation of the caller/callee relationships between functions without requiring any source code changes.

Figure 3 shows a sample call graph. Functions are color-coded to show which ones take the most time, and tooltips show details about the function and call relationships.



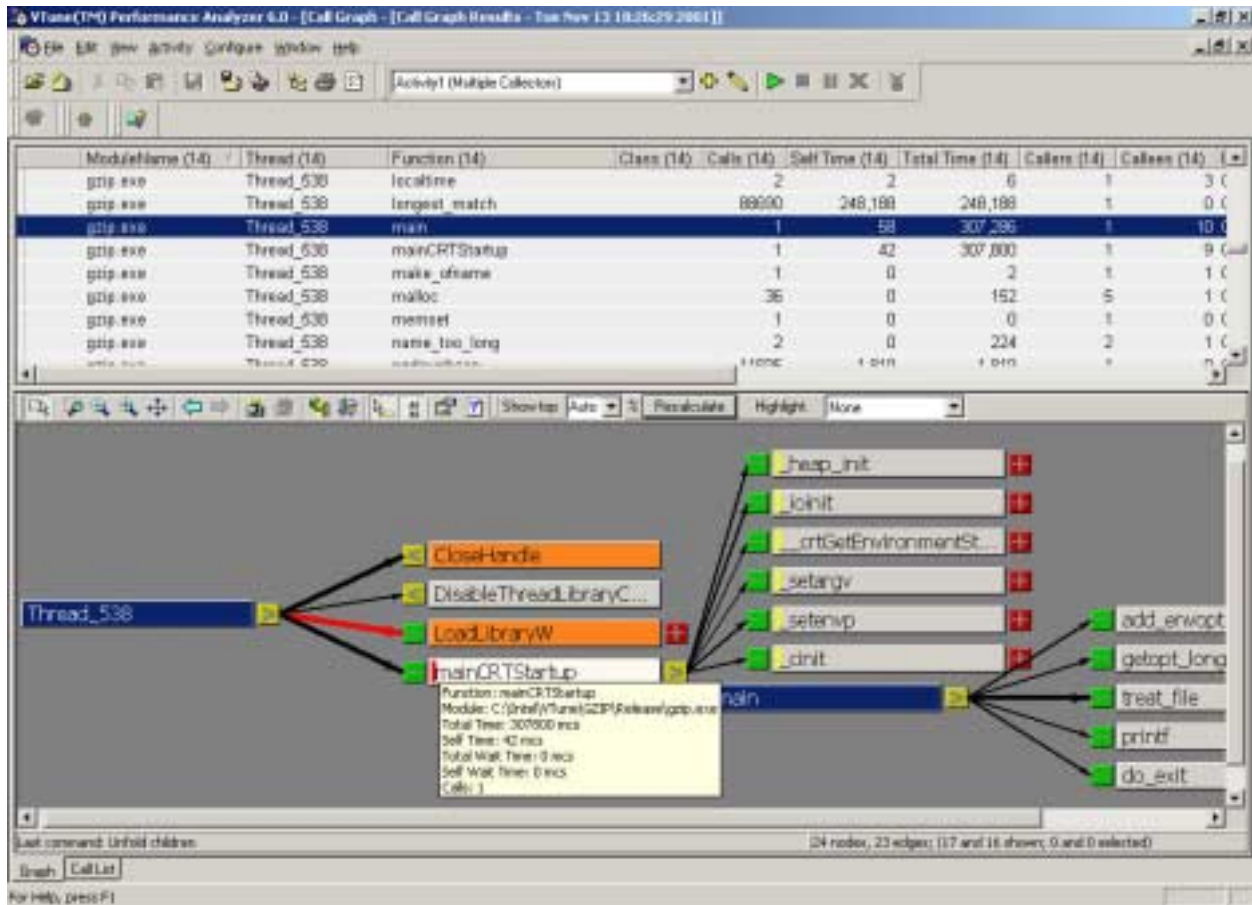


Figure 3. The call graph view enables you to clearly see what functions call others, and which ones use the most time

In many cases, a function may be called from several different places. The call graph collector can provide call information per site.

### Monitoring Performance Counter Activity

Sometimes understanding an application’s performance requires a higher-level perspective. The VTune Performance Analyzer can monitor hardware and software counters<sup>3</sup>. This feature enables you to track system activity during runtime, which helps you understand the cause-and-effect relationship between the computer’s subsystems and your application. This perspective on performance can indicate when an application may be exhibiting anomalous behavior and allows you to focus only on those times when the behavior is occurring.

By default, the Quick Performance Analyzer (QPA) Wizard monitors the following set of counters:

- Redirector: Writes Denied/sec
- Redirector: Reads Denied/sec
- Redirector: Current Commands
- Redirector: Bytes Total/sec

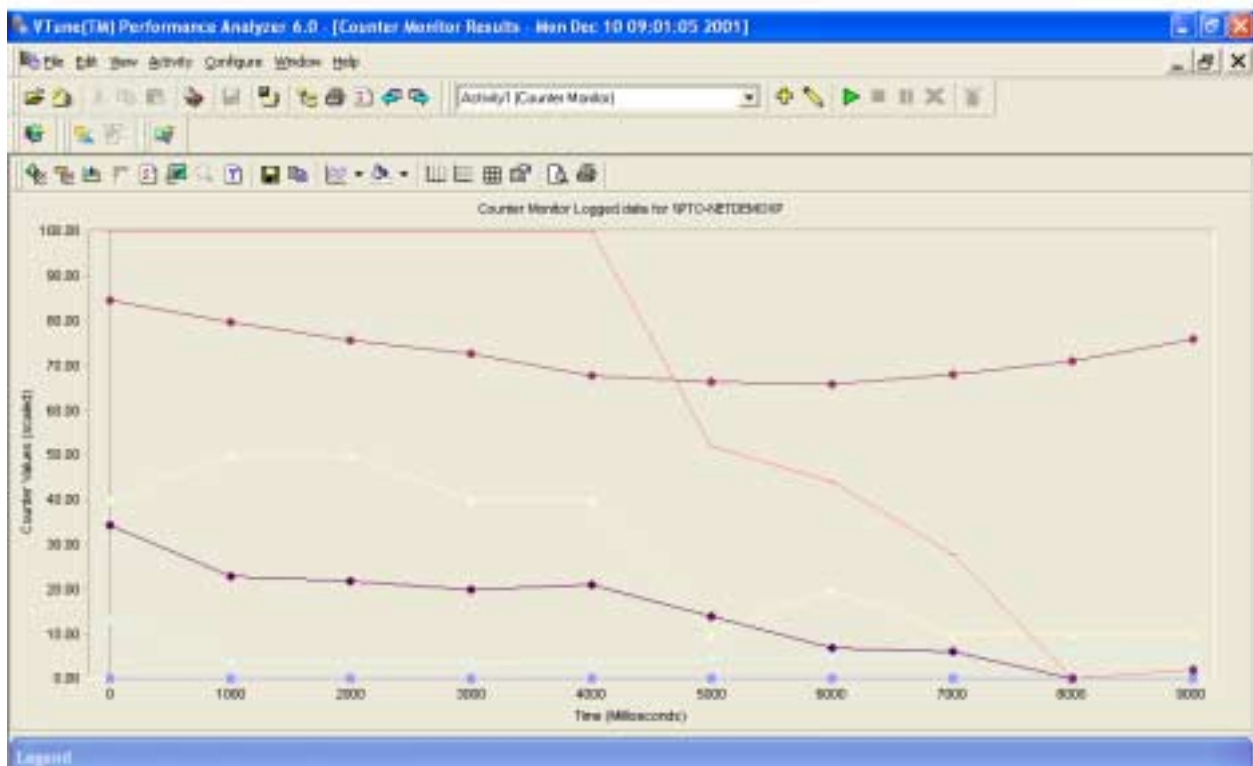
<sup>3</sup> If you develop application-specific counters using Performance DLLs, the VTune Performance Analyzer will also collect and display these counters.

## Introduction to the Intel® VTune™ Performance Analyzer

- Redirector: Network Errors/sec
- Memory: Pages/sec
- Memory: Available Bytes
- System: Processor Queue Length
- System: Context Switches/sec
- Processor (\_Total): % Privileged Time
- Processor (\_Total): % Processor Time

The VTune Performance Analyzer's extensive context-sensitive help system explains what each counter measures.

Figure 4 shows a sample counter monitor view.



**Figure 4.** The counter monitor view shows how your application interacts with the system resources

### ***Suggesting Code Modifications***

The VTune Performance Analyzer comes with a Tuning Assistant, which can examine how your application interacts with the system and can provide context-sensitive tuning advice that may improve your code. Figure 5 shows sample Tuning Assistant output.

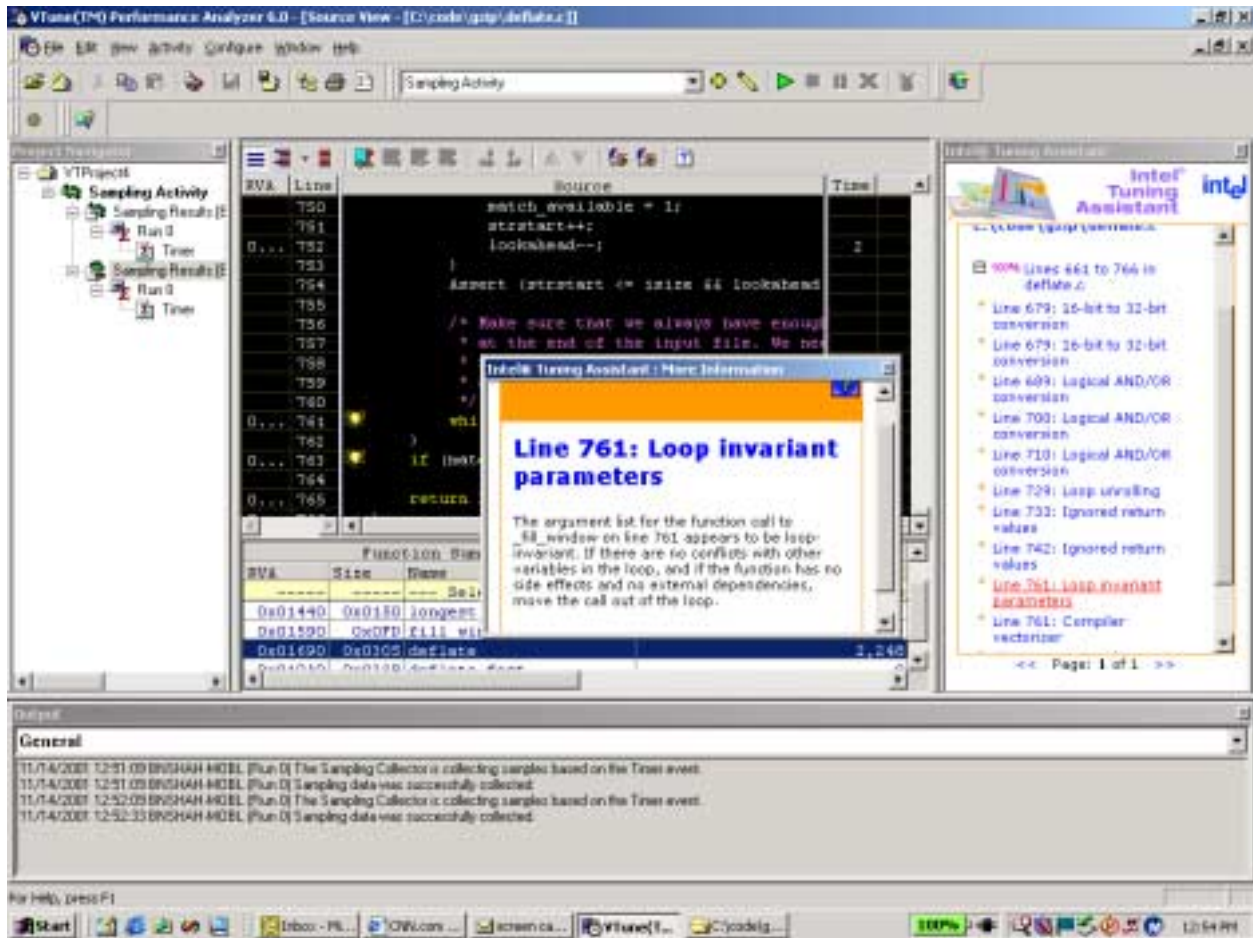


Figure 5. The Tuning Assistant offers succinct and specific advice about your code

## Obtaining the Intel® VTune™ Performance Analyzer

The Intel VTune Performance Analyzer can help you identify performance issues in software that runs on Intel processors. Included with the VTune Performance Analyzer is a tutorial that covers basic concepts and walks you through an example Quick Performance Analysis using the QPA wizard. An extensive online help system completes the package.

For more information about how to get started using the VTune Performance Analyzer, visit <http://www.developer.intel.com/products/software/vtune>. In particular, the Web site offers detailed information about which operating systems and languages are supported. If you are not yet convinced that the VTune Performance Analyzer can help you improve your code, take advantage of the 30-day free trial (see the Web site for more details).