



Using the VTune Analyzer on Multithreaded Applications

Intel Software College



Objectives

Two uses for VTune Performance Analyzer

- Improving the efficiency of computation
- Improving your threading model

Example: mandelbrot3.exe



2

Using the VTune Analyzer on Multithreaded Applications

Copyright © 2005, Intel Corporation. All rights reserved.

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States or other countries. Other brands and names are the property of their respective owners.



Agenda

Determine if a sample section of code can be threaded

Determine if the implemented threads are balanced or not



3

Using the VTune Analyzer on Multithreaded Applications

Copyright © 2005, Intel Corporation. All rights reserved.

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States or other countries. Other brands and names are the property of their respective owners.



Use VTune Analyzer on Serial Applications

Determine what parts of your application (if any) that, when threaded, will speed up your app

- CPU-bound apps can potentially run twice as fast on dual-core processors
- Memory bound apps may potentially run 50% faster
- I/O bound applications may not run any faster

Find main performance bottlenecks in your application (sampling)

- Determine whether or not it makes sense to thread there
- If not, look further up the program's calling sequence to find a more appropriate place to consider (callgraph)



4

Using the VTune Analyzer on Multithreaded Applications

Copyright © 2005, Intel Corporation. All rights reserved.

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States or other countries. Other brands and names are the property of their respective owners.



Use VTune Analyzer on Multithreaded Applications

Determine if your current threading model is balanced

- On the thread view, each of the threads should be consuming the same amount of time
- If not, you must shift the amount of work between the threads
- Use "Samples Over Time" view

Look for idle CPU time

- Could more threads utilize idle resources?
- In the process view this can show up as ntoskrnl, or intelppm.sys
- Sometimes the process view actually says idle task



Example: mandelbrot3.exe

Single threaded app, analyze to see if it can be made

Demo machine:

- Four socket, dual core, hyper-threaded (HT) machine
- (4 sockets * 2 cores per socket * 2 HT CPUs per core) = the OS sees 16 processors





