Using Intel® VTune[™] Amplifier XE for High Performance Computing

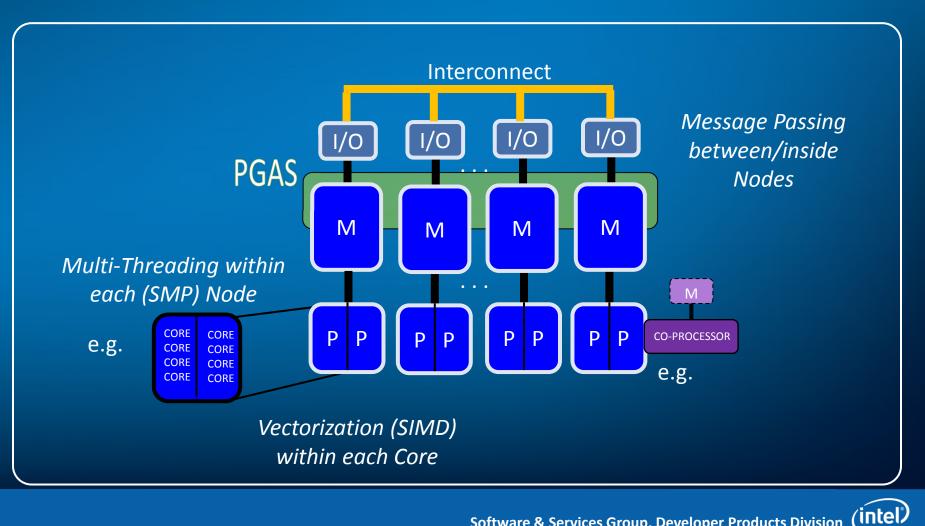
Vladimir Tsymbal Performance, Analysis and Threading Lab



Software & Services Group, Developer Products Division

Copyright© 2010, Intel Corporation. All rights reserved. *Other brands and names are the property of their respective owners. Software

The Majority of all HPC-Systems are Clusters



Software & Services Group, Developer Products Division



Copyright© 2010, Intel Corporation. All rights reserved. *Other brands and names are the property of their respective owners.

Common Tools and Programming Models

Multicore

Intel® Xeon® processors are designed for intelligent performance and smart energy efficiency



Continuing to advance Intel® Xeon® processor family and instruction set (e.g., Intel® AVX. etc.)





Software development platforms ramping now

Use One Software Architecture Today. Scale Forward Tomorrow.



High Performance Software Products Supporting Multicore and Many-core Development

Intel® Parallel Studio XE* Advanced Performance



Intel® C/C++ and Fortran Compilers w/OpenMP

Intel® MKL, Intel® Cilk Plus, Intel® TBB Library, Intel® ArBB Library Intel® IPP Library

Intel® Inspector XE, **Intel**® VTune[™] Amplifier XE, Intel® Advisor



Intel® Cluster Studio XE* Distributed Performance

Intel[®] MPI Library

Intel[®] Trace Analyzer

Intel[®] Parallel Studio XF

Performance. Scale Forward. Proven

Software & Services Group, Developer Products Division



Optimization Notice

Copyright© 2010, Intel Corporation. All rights reserved. *Other brands and names are the property of their respective owners. Software

MPI Analysis VTune Amplifier XE and MPI interoperability

- To analyze the performance and correctness of an MPI program at the <u>inter-process</u> level
 - use the Intel® Trace Analyzer and Collector
 - collects information about events at the MPI level between processes
 - allows analyzing the performance and correctness of the MPI calls, deadlock detection, data layout errors, as well as risky or incorrect MPI constructs
 - The Intel Trace Analyzer and Collector data is correlated and aggregated across all processes and all nodes, that participated in the execution run



MPI Analysis (ITAC) Hybrid program: 2 MPI processes + 12 Threads per process

🚟 intel® Trace /	nalyzer - [1: C:/	Documents an	d Settings/hbock	or/My Documents/	Projects/AXE/Prese	ntation Dresden/Re	esults/ITAC/bqcd4	4.single.stf]				3
📴 <u>F</u> ile <u>P</u> roject <u>S</u>	yle <u>W</u> indows <u>H</u> e	lp F1										- 8 ×
<u>V</u> iew <u>C</u> harts <u>N</u> av	igate <u>A</u> dvanced											
4.36 s	4.38 s	4.40 s	4.42 s	4.44 s	4.4 4.46 s	3 s 4.50 s	4.52 s	4.54 s	4.56 s	4.58 s	4.60 s	~ ~
Ро <mark>ия аданный канал</mark>	WWWMP1_Allreduce						A.1976	KURUUPAJAJAJAJAJAJAJA	Applic/Applic///////////////////////////////////	ЧЧЧАррі «ЧАр/Ар/Ар/	\. \	آم «App»
Р1 <i>///Ар.Р/Л-И/Л-И/Л-И</i>	www.Application					MPI_Alfred	fuce 454		Applic -Applic w. +hm.	анул-Аррик-ЗАріАри	Аррини Ариари Али	тАррия
<												>
Flat Profile Load	Balance Call Tree	Call Graph										
Group All_Processes	~											
	Ication 329.353e Ice 175.976e 11.493e 4.211e I 3.834e 2.221e Ize 914e	-3 s -3 s -3 s -3 s -3 s	528.03e-3 s 175.976e-3 s 11.493e-3 s 4.211e-3 s 3.834e-3 s 2.221e-3 s 914e-6 s	<pre>#Calls TSelf /Call 0</pre>								
	4.351 537, 4.615 55	2: 0.264 015		ec.	All_Proce	2522		MPI expander	d in (Major Function G	iroups)	Tag	Filter
		2. 0.204 015			ALTICA		re & Serv	ices Grou				



MPI Analysis VTune Amplifier XE and MPI interoperability

- Hybrid programming
 - Beyond the inter-process level of MPI parallelism, the processes that make up the programs on a modern cluster often also use fork-join threading through OpenMP* and Intel® TBB
 - VTune Amplifier XE performance analyzer and the Intel Inspector XE checker can be used to analyze the performance and correctness of an MPI program

Software & Services Group, Developer Products Division



Optimization Notice Copyright© 2010, Intel Corporation. All rights reserved. *Other brands and names are the property of their respective owners. Software

Hybrid Analysis in 2 steps

1. Use the 'amplxe-cl' command line tools to collect data

- By default, all processes are analyzed, but it is possible to filter the data collection to limit it to a subset of processes.
- An individual result directory is created for each spawned MPI program process that was analyzed with MPI process rank value captured.
- Post-processing, also called "finalization" or "symbol resolution", is done automatically for each result directory once the collection has finished.

2. Open and analyze each result directory through the GUI standalone viewer



Hybrid Analysis Collecting Performance Data

\$ mpirun -n <N> amplxe-cl -r my_result -collect <analysis type>
my_app [my_app_ options]

where <abbr> is "amplxe" or "inspxe"

- The list of analysis types available can be viewed using "amplxe-cl help collect" command.
- For hotspots: an example of full command line for collection would be:

\$ mpirun -n 4 amplxe-cl -r my_result -collect hotspots -- my_app 12

 Number of result directories will be created in the current directory, named as my_result.0 – my_result.3



Hybrid Analysis Collecting Performance Data

- Sometimes it is necessary to collect data for a subset of the MPI processes in the workload.
- Example: 16 processes in the job distributed across the hosts and hotspots data should be collected for only two of them:
- \$ mpirun -host myhost -n 14 ./a.out : -host myhost -n 2 amplxe-cl -r foo -c hotspots ./a.out
 - 2 directories will be created in the current directory: foo.14 and foo.15 (given that process ranks 14 and 15 were assigned to the last 2 processes in the job).



Hybrid Analysis Finalizing the Collected Data

- Finalization of the data (symbol resolution, conversion to the database) happens automatically after the collection has finished
 - happens on the same compute node where the command line collection was executing
 - Binaries and symbol files will be located automatically
 - In cases where the search algorithm needs to be adjusted (common reason: need to point to symbol files stored elsewhere), the <u>-search-dir</u> option should be used with amplxe-cl as follows:

\$ mpirun -np 128 amplxe-cl -q -collect hotspots -searchdir sym=/home/foo/syms ./a.out



Hybrid Analysis Viewing the Collected Data

- Once the results are collected, the user can open any of them in the standalone GUI or generate a command line report.
 - Use amplxe-cl –help report to see the options available for generating reports.
 - Here is an example of viewing the text report for functions and modules after a VTune Amplifier XE analysis:

\$ amplxe-cl -R hotspots -q -format text -r r003hs

Function Module CPU Time

F Main	a.out a.out	6.070 2.990
\$ amplxe-cl -R Module		<mark>s -q -format text -group-by module –r r003hs</mark> me
a.out	9.060	

Software & Services Group, Developer Products Division

intel

Hybrid Analysis Visualize results in VTune™ Amplifier XE

- Linux: start
- \$ amplxe-gui
- Open results
 r003hs.amplxe

2	<no current="" project=""> - Intel VTune Amplifier XE 2011</no>						
<u>F</u> ile Help							
🛯 🔛 🖆 🖄 📄 🕑 🚅 🕕							
Welcome to Intel VTu	ine Amplifier XE 2011!						
Open an Intel VTune Amplif following options:	fier XE 2011 project to proceed. Use any of the						
* Create a new project:	File > New > Project						
* Do one of the following	g to open an existing project:						
File > Open > Proj	File > Open > Project						
File > Recent Proje	ects						
* Do one of the following opens the corresponding p	g to open a previously collected result (which roject):						
File > Open > Rest	ult						
File > Recent Resu	lts						
Alternatively, you can com	pare results: File > Open > Compare Results						
Tip: Open an analysis resul to graphically visualize the	It collected with the amplxe-cl command-line tool performance data.						

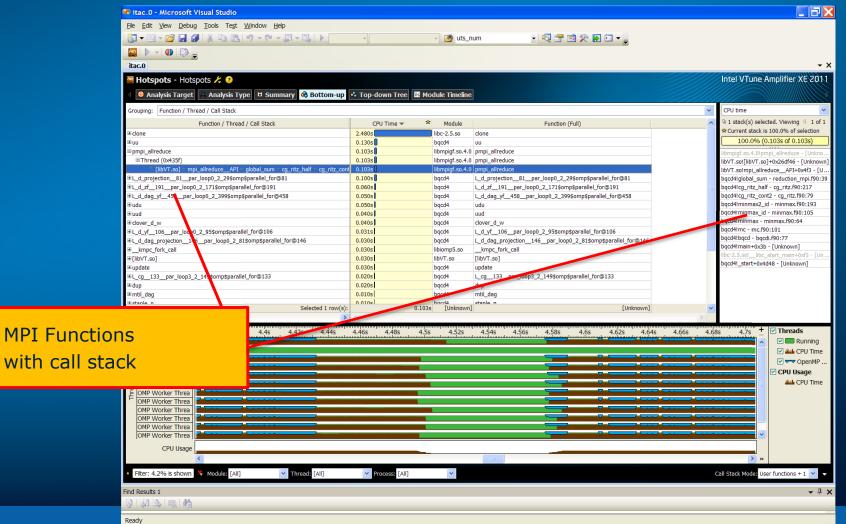


Hybrid Analysis (hotspots) Hybrid program: 2 MPI processes + 12 Threads per process

🍩 itac.0 - Microsoft Visual Studio				
<u>Elle Edit View Debug T</u> ools Te <u>s</u> t <u>W</u> indow	Help			
		vits_num	- 🔩 🚰 🐋 🛠 🛃 🖸 🗕	
🦉 I 🕑 🚽 🕩 💭 📮				
itac.0				
💹 Hotspots - Hotspots 🗶 🥺				Intel VTune Amplifier XE 2
d 🔮 Analysis Target 🔺 Analysis Type 🕅 St	Summary 🚳 Bottom-up 🗳 Top-down Tree 🗄	Module Timeline		
Grouping: OpenMP Regions / Function / Call Stack				Task stack
OpenMP Regions / Function / Call Stack	all Stack CPU Time 🔻	* Module Function (Full)		A 1 stack(s) selected. Viewing 4 1
[Outside any OpenMP parallel region]	53.089s	Module Function (Full)		* Current stack is 100.0% of selecti
OMP Parallel-clover_d_loop_w_:210-236	3.753s			100.0% (53.089s of 53.089s
⊞uu	1.2295	bqcd4 uu		[Unknown]
⊞udu ⊞uud	0.760s	bqcd4 udu bqcd4 uud		
■ uuu ■ L_clover_d_loop_w210par_loop1_2_192\$omp5		bqcd4 L_clover_d_loop		-
□kmp_wait_sleep	0.190s	libiomp5kmp_wait_sl		
⊞u_sub	0.170s	bqcd4 u_sub		
Bkmp_launch_thread B clover_d_same_eo_w	0.1485	libiomp5kmp_launch bqcd4 clover_d_same		
■ clover_d_diff_eo_w	0.101s	bqcd4 clover_d_diff_e		
■_kmp_x86_pause	0.075s	libiomp5kmp_x86_pa		
clover_d_loop_w	0.070s	bqcd4 clover_d_loop_w libiomp5kmp_execute		
MD Decience	0.000s	libc-2.5.sosched_yield		
nMP Regions	0.010s	[stack] [[stack]]		
	3.609s 3.211s			
routines called	1.870s			
		.089s [Unknown] [Unknown]		v
le 📃	> <			>
1s	1.5s 2s 2.5s 3s	3.5s 4s 4.5s	5s 5.5s 6s 6.5s 7s 7	.5s 8s 8.5 + Threads
Thread (0x435f)				Runni
Thread (0x4376) OMP Worker Threa				CPU T
OMP Worker Threa				CPU Usage
OMP Worker Threa				CPU USAGE
OMP Worker Threa				
OMP Worker Threa				
OMP Worker Threa				
OMP Worker Threa				
OMP Worker Threa				
OMP Worker Threa				
OMP Worker Threa				·····································
OMP Worker Threa	✓ Thread: [All] ✓ Process: [All]			Call Stack Mode: Only user functions
OMP Worker Threa OMP Worker Threa CPU Usage	✓ Thread: [All] ✓ Process: [All]			Call Stack Mode: Only user functions
OMP Worker Threa OMP Worker Threa OMP Worker Threa CPU Usage				Call Stack Mode: Only user functions
MP Worker Threa OMP Worker Threa OMP Worker Threa CPU Usage CPU Usage A No filters are appled. A Module: [All] Find Results 1 COncen MP				Call Stack Mode: Only user functions
MP Worker Threa MP Worker Threa CPU Usage CPU Usage Find Results 1 Pind Results 1 CPU Bage CPU Bage	threads shown			Call Stack Mode [®] Only user functions
MP Worker Threa MP Worker Threa CMP Worker Threa CPU Usage CPU CPU Usage CPU CPU CPU CPU CPU CPU CPU CPU CPU CPU	threads shown			•
MP Worker Threa Modules CPU Usage CPU Usage Mo filters are appled. * Modules [All] Find Results 1 []			tware & Services Group, D	•
MP Worker Threa MP Worker Threa CPU Usage CPU Usage Find Results 1 Ready Ready OpenMP together	threads shown) Soft		Developer Products
MP Worker Threa OMP Worker Threa OMP Worker Threa CPU Usage ✓ No fitters are appled. ★ Modules [All] Find Results 1 Prot Results 1 Ready CPU Bage	threads shown) Soft	tware & Services Group, D	Developer Products

Op wit ins

MPI Analysis (hotspots) Hybrid program: 2 MPI processes + 12 Threads per process





Summary

- VTune[™] Amplifier XE is the coming together of distinct tools
 - Correlation of the data from various capabilities makes it very powerful
 - Supports EBS data collection with better usability
 - Supports statistical call-graph capability
 - Combines these two types of capabilities with powerful Thread Profiling capability
- VTune[™] Amplifier XE provides a standalone GUI with the same look and feel on both Windows* and Linux
- Complete re-design to make it a very extensible tool

(intel)



Software



Optimization Notice

Optimization Notice

Intel compilers, associated libraries and associated development tools may include or utilize options that optimize for instruction sets that are available in both Intel and non-Intel microprocessors (for example SIMD instruction sets), but do not optimize equally for non-Intel microprocessors. In addition, certain compiler options for Intel compilers, including some that are not specific to Intel micro-architecture, are reserved for Intel microprocessors. For a detailed description of Intel compiler options, including the instruction sets and specific microprocessors they implicate, please refer to the "Intel Compiler User and Reference Guides" under "Compiler Options." Many library routines that are part of Intel compiler products are more highly optimized for Intel microprocessors than for other microprocessors. While the compilers and libraries in Intel compiler products offer optimizations for both Intel and Intel-compatible microprocessors, depending on the options you select, your code and other factors, you likely will get extra performance on Intel microprocessors.

Intel compilers, associated libraries and associated development tools 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 Intel® Streaming SIMD Extensions 2 (Intel® SSE2), Intel® Streaming SIMD Extensions 3 (Intel® SSE3), and Supplemental Streaming SIMD Extensions 3 (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.

While Intel believes our compilers and libraries are excellent choices to assist in obtaining the best performance on Intel and non-Intel microprocessors, Intel recommends that you evaluate other compilers and libraries to determine which best meet your requirements. We hope to win your business by striving to offer the best performance of any compiler or library; please let us know if you find we do not.

Notice revision #20110307



Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, reference www.intel.com/software/products.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

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

Copyright ©2010. Intel Corporation.

http://www.intel.com/software/products



Software & Services Group, Developer Products Division

Optimization Notice Copyright© 2010, Intel Corporation. All rights reserved. *Other brands and names are the property of their respective owners. Software