

Intel Tuning for Juwels and Jureca

FZ-Jülich, November 23, 2023 Dr. Heinrich Bockhorst - Intel Heinrich.Bockhorst@Intel.com



class MirrorX(bys.types.t

(classected) (class control)) (class control))

Agenda

- oneAPI Initiative
- Intel[®] Compiler
- Application Performance Snapshot (APS)
- VTune Profiler
- Advisor
- Intel[®] MPI

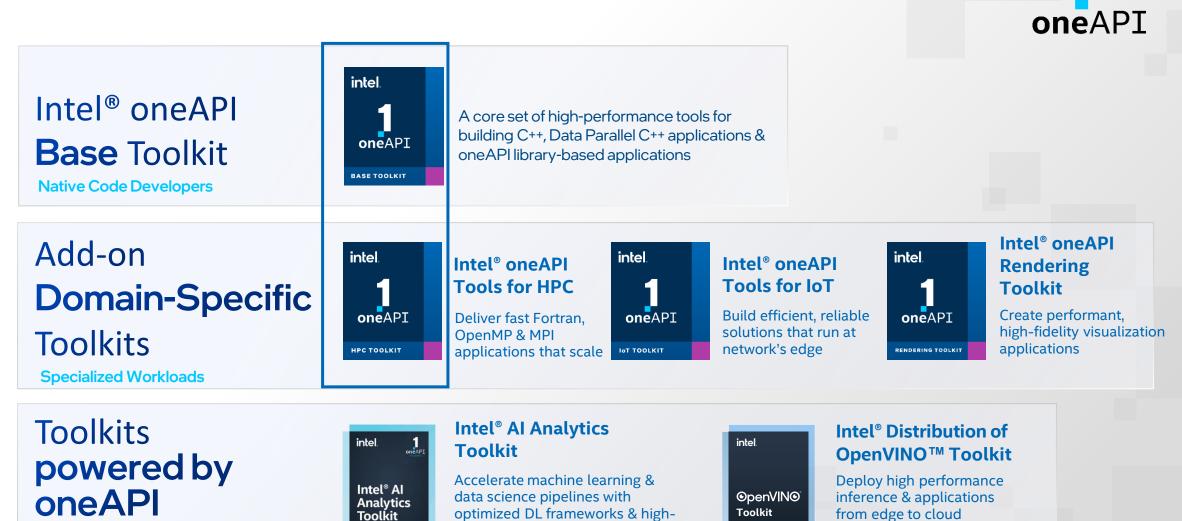
Cross-Architecture Programming for Accelerated Compute, Freedom of Choice for Hardware ONEAPI: Industry Initiative & Intel Products

One Intel Software & Architecture group Intel Architecture, Graphics & Software November 2020





Free Download of all packages!



performing Python libraries

Data Scientists & Al Developers

Get started quickly Code Samples, Quick-start Guides, Webinars, Training <u>https://software.intel.com/oneapi</u> https://cloud.intel.com



New Cloud Access cloud.intel.com

Select (free access)

"Training and Workshops"

- "Hardware Catalog" shows more options only for paying customers!
- Download Samples from:

https://github.com/oneapi-src/oneAPI-samples

Quick Start	Learning and Support	Notifications	
Hardware Catalog	Getting started Learn the fundamentals to get the Most out of the Intel developer cloud	\cap	
Training and Workshops Cloud Credits	Tutorials Browse how to create better	No notifications yet	
Cidud Credits.	solutions using Intel developer cloud	Stay tuned for exciting updates! No new notifications at the moment.	
	Learn the fundamentals to get the Most out of the Intel developer cloud		
Gen Al Essentials			
Text-to-Image with Stable Diffusion A Creative Playeround for Artists, W	Image-to-Image Generation with Stable Diffusion Perfect for artists and engineers who want	Simple LLM Inference: Playing with Language Models A hands-on experience on language	
Launch	to see their images transform in creative and unexpected ways.	models and text generation, no technical background needed.	
Company Overview Contact Intel	Newsroom Investors Careers Corporate Responsibility Diversit	y & Inclusion Public Policy	
ntel 6 y 🖬 🔍 🞯			

New Cloud Access cloud.intel.com

- Free Access to new Intel CPU/GPU hardware
- Training material on Jupyter notebooks.

	DS	
AI		
AI Kit XGBoost Predictive Modeling Learn predictive modeling with decision trees using Intel [®] AI Analytics Toolkit	Heterogeneous Programming Using Data Parallel Extension for Numba [®] for AI and HPC Data Parallel Extension for Numba accelerates Python® code on Intel [®] XPUs	Machine Learning Using oneAPI Intel [®] AI Analytics Toolkit accelerates data science and analytics with Python [®]
😴 Launch 🖸	₩ Launch	😸 Launch 🖸
C++ SYCL		
Essentials of SYCL Learn to write performant and portable code using oneAPI and SYCL C++	Performance, Portability and Productivity Learn to write performant and portable HPC code for multiple platforms with oneAPI and SYCL C++	Introduction to GPU Optimization Learn GPU optimization techniques using SYCL.
j Launch ☑	🛱 Launch 🖾	≓ Launch 🗹
Migrate from CUDA® to C++ with SYCL® Optimize apps from traditional CUDA environments		
👼 Launch 🖾		
Gen Al Essentials		
Text-to-Image with Stable Diffusion A Creative Playground for Artists, Writers, and Engineers	Image-to-Image Generation with Stable Diffusion Perfect for artists and engineers who want to see their images transform in creative and unexpected ways.	Simple LLM Inference: Playing with Language Models A hands-on experience on language models and text generation, no technical background needed.
避 Launch 🖾	≓ Launch 🗗	≓ Launch 🗹



Intel[®] Compilers

Intel[®] Compilers Going Forward

New underlying back-end Compilation Technology based on LLVM

New compiler technology available today in Intel® oneAPI Base & HPC Toolkit for DPC++, C++ and Fortran

Existing Intel proprietary "ILO" (ICC, IFORT) Compilation Technology compilers provided alongside new compilers

CHOICE! Continuity!

BUT Offload (DPC++ or OpenMP TARGET) supported only with new LLVM-based compilers

All Intel compilers are available on Juwels/Jureca: \$ module load Intel

Intel[®] Compilers

Intel Compiler	Target	OpenMP Support	OpenMP Offload Support	Included in oneAPI Toolkit
Intel [®] C++ Compiler Classic, ILO <i>icc/icpc/icl</i>	CPU	Yes	No	HPC
Intel [®] Fortran Compiler Classic, ILO <i>ifort</i>	CPU	Yes	No	HPC
Intel® Fortran Compiler, LLVM <i>ifx</i>	CPU, GPU	Yes	Yes	HPC
Intel® oneAPI DPC++/C++ Compiler, LLVM <i>dpcpp</i>	CPU, GPU, FPGA*	Yes	Yes	Base
Intel® oneAPI DPC++/C++ Compiler, LLVM <i>icx/icpx</i>	CPU GPU*	Yes	Yes	Base

Compiler Binary Compatible and Linkable!

tinyurl.com/oneapi-standalone-components

22-2025

Compiler	XPU Support		Compiler Status/Maturity Schedule								
Intel® C++ Compiler Classic	CPU	2022 Q1Q2Q3Q4Production QualityDeprecation announced	2023 Q1 Q2 Q3 Q4 LPS Removed from oneAPI toolkits	2024 Q1 Q2 Q3 Q4 2025 Q1 Q2 Q3 Q4	 Not recommended for new projects Start/continue migration now Last supported architecture: Sapphire Rapids 						
Intel® oneAPI DPC++/C++ Compiler	CPU GPU FPGA	Production Quality Production Quality Production Quality			• Use for all new projects						
Intel® Fortran Compiler Classic	CPU	Production Quality		LPS#	 Prepare to migrate in 2022 Last supported architecture: Sapphire Rapids #LPS start date TBD 						
Intel® Fortran Compiler	CPU GPU	Production Quality Production Quality	Est. feature/perf. parity with Clas	asic	 Test drive now & provide feedback 						

CPU = Intel[®] Xeon[®] and Core[™] processors GPU = Intel[®] Integrated and discrete GPU's FPGA = Intel[®] FPGA's (Stratix, Arria, Agilex)

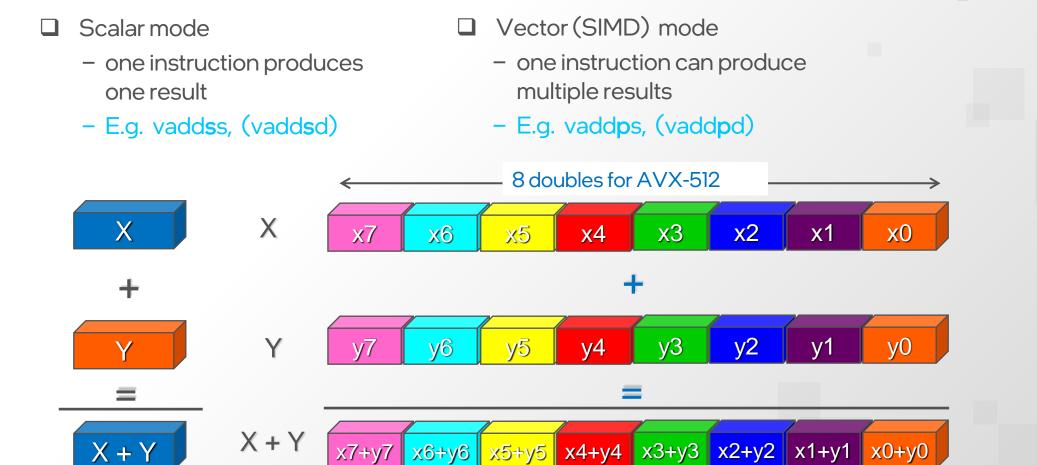
LPS = Legacy Product Support

Common optimization options

	Linux* icx (icc)
Disable optimization	-00
Optimize for speed (no code size increase)	-01
Optimize for speed (default)	-02
High-level loop optimization	-03
Create symbols for debugging	-g
Multi-file inter-procedural optimization	-ipo
Profile guided optimization (multi-step build)	-fprofile-generate (-prof-gen) -fprofile-use (-prof-use)
Optimize for speed across the entire program ("prototype switch")	-fast same as "-ipo -O3 -static -fp-model fast" (-ipo -O3 -no-prec-div –static -fp-model fast=2 -xHost)
OpenMP support	-fiopenmp (-qopenmp)

SIMD: <u>Single Instruction</u>, <u>Multiple Data</u>

for (i=0; i<n; i++) z[i] = x[i] + y[i];</pre>



Basic Vectorization Switches I

- Linux*, OS X*: -x<feature>
 - Might enable Intel processor specific optimizations
 - Processor-check added to "main" routine: Application errors in case SIMD feature missing or non-Intel processor with appropriate/informative message
 - Example: -xCORE-AVX512 (Juwels Xeon SKL)
- Linux*, OS X*: -ax<features>
 - Multiple code paths: baseline and optimized/processor-specific
 - Multiple SIMD features/paths possible, e.g.: -axSSE2, CORE-AVX512
 - Baseline code path defaults to -xSSE2

Basic Vectorization Switches II

- Special switch for icc, Linux*, OS X*: -xHost
- Compiler checks SIMD features of current host processor (where built on) and makes use of latest SIMD feature available
- Code only executes on processors with same SIMD feature or later as on build host



LLVM-BASED INTEL COMPILERS

What is ICX?

- Close collaboration with Clang*/LLVM* community
- ICX is Clang front-end (FE), LLVM infrastructure
 - PLUS Intel proprietary optimizations and code generation
- Clang FE pulled down frequently from open source, kept current
 - Always up to date in ICX
 - We contribute! Pushing enhancements to both Clang and LLVM
- Enhancements working with community better vectorization, opt-report, for example

tinyurl.com/blog-on-icx

Major Changes Overview <u>tinyurl.com/icc-to-icx-migration-guide</u>

- LLVM is a different compilation technology. EXPECT differences
- Options:
 - icx -qnextgen-diag option to get a list of supported and unsupported options
- Use-fiopenmpor-fiopenmp-simd for OpenMP
- C/C++ Pragmas a lot of Intel proprietary ones not supported
 - enable -Wunknown-pragmas to warn on unsupported pragmas
- INTEL_LLVM_COMPILER is defined instead of __INTEL_COMPILER

Please switch to icx/icpx Compiler!

- Deprecation planed for 2024
- Check the user guide for supported flags:

https://www.intel.com/content/www/us/en/docs/dpcpp-cppcompiler/developer-guide-reference/2024-0/overview.html

- Check results and compare with icc/icpc results:
 - -fp-model=fast is the default
 - -fp-model=precise might help to reproduce previous results

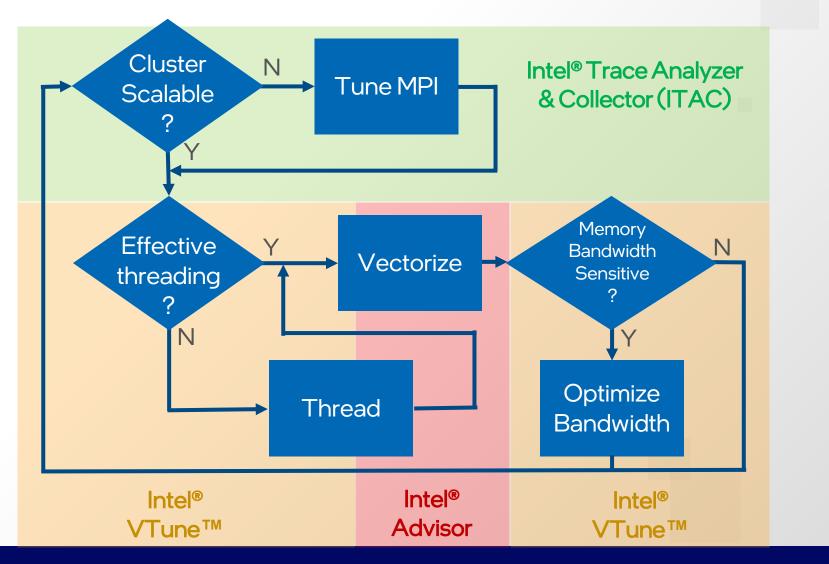
Build your own compiler (only for experts)

- Most of the features are included in the public llvm Intel version. You may test and contribute to the development.
- Interested? Check out: <u>https://intel.github.io/Ilvm-docs/GetStartedGuide.html</u>
- Build a clang compiler with latest features ahead of icpx and icx
- Some features may be missing in the public version
- Can also configure and build a CUDA/AMD backend compiler for offload to NVIDIA/AMD cards
- NVIDIA backend is also available for the oneAPI icpx version: <u>https://developer.codeplay.com/products/oneapi/nvidia/2024.0.0/guides/get-started-guide-nvidia.html</u>



Which tool should I use?

Performance Analysis Tools for Diagnosis



Before dive to a particular tool..

- How to assess easily any potential in performance tuning?
- What to use on big scale not be overwhelmed with huge trace size, post processing time and collection overhead?
- Which tool should I use first?
- Answer: try Application Performance Snapshot (APS)
- Look for VTune module if available

APS Usage

Setup Environment

\$ source <path_to_vtune>/vtune_vars.sh # or load module

Run Application

- \$ aps <application and args>
- MPI: \$ mpirun < mpi options > aps < application and args >

Generate Report on Result Folder

\$ aps -report <result folder>



Generate CL reports with detailed MPI statistics on Result Folder

\$ aps-report -<option> <result folder>

	Volume (NB)		
Rank> Rank	vor abe (mb)	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	IL anovero
0023> 0024	84.35		
0025> 0026	84.35		
0024> 0025	84.15		
0021> 0022	83.84		
[filtered out 1	6 lines]		
	69.60		
	68.78		
	68.38		
	58.81		
	56.98		
[filtered out 1			

Application Performance Snapshot (APS)

Data in One Place: MPI+OpenMP+Memory Floating Point

Quick & Easy Performance Overview

- Does the app need performance tuning?
- MPI & non-MPI Apps⁺
- Distributed MPI with or without threading
- Shared memory applications

Popular MPI Implementations Supported

- Intel[®] MPI Library
- MPICH & Cray MPI

Richer Metrics on Computation Efficiency

- CPU (processor stalls, memory access)
- FPU (vectorization metrics)

Application: my_app Report creation date: 2017-1 Number of ranks: 4 OpenMP threads per rank: 2 HW Platform: Intel(R) Xeon(R Logical Core Count per node 21.075	2 1) Processor cod	e named Broadwell-EP	imbalance.	-	ificant OpenMP ▶ <u>VTune[™] Amplifier</u> to see the Reta		
Elapsed Time			MPI Time OpenMP Imbalance Memory Stalls	2.84% <15% 36.40% ► <10% 32.98% ► <20%			
226.07	1.16		FPU Utilization	3.12%► >50%			
SP FLOPS	CPI		I/O Bound	0.00% <10%			
MPI Time 2.84% of Elapsed Time (0.60s) MPI Imbalance 1.83% of Elapsed T (0.39s №) TOP 5 MPI Functions Waitall Irecv		OpenMP Imbalance 36.40% of Elapsed Time (7.67s) Memory Footprint Per node: Pegk; 7125.32 MB Average: 7125.32 MB Per rank:	32.98% of p Cache Sta 21.63% DRAM.Sta 1.50% of NUMA	oipeline slots I <u>lls</u> of cycles a lls	FPU Utilization 3.12% SP.FLOPs per Cycle 1.00 Out of 32.00 Vector Capacity Usage 33.10% FP Instruction Mix % of Packed FP. Instr:		
Barrier	0.11	<u>Peak:</u> 1795.30 MB Average: 1781.33 MB			10.80% % of <u>128-bit</u> : 0.00%		
Reduce	0.01	Average. 1761.55 Mb			% of <u>256-bit</u> : 10.80%		
lsend	0.01				% of <u>Scalar FP Instr.</u> : 89.20		
I/O Bound 0.00% (AVG 0.00, PEAK 0.00) Read AVG 0.0 KB, MAX 0 Write	0.0 KB				FP Arith/Mem Rd Instr. Ratio 0.98 FP Arith/Mem Wr Instr. Ratio 2.47		

⁺MPI supported only on Linux^{*}

APS Command Line Reports – Advanced MPI statistics

 Data Transfers for Rankto-Rank Communication
 aps-report –x <result>

And many others – check • aps-report -help

	Volume(MB)	Volume(%)	Transfers
1			
	84.35		13477
0025> 0026	84.35	1.56	13477
0024> 0025	84.15	1.56	13477
0021> 0022	83.84	1.55	13477
0022> 0023	83.43	1.54	13477
[filtered out	16 lines]		
0012> 0011	69.60	1.29	13477
0020> 0019	69.29	1.28	13477
0026> 0025	68.78	1.27	13477
0025> 0024	68.38	1.27	13477
0022> 0021	68.38	1.27	13477
[filtered out	17 lines]		
0016> 0015	58.81	1.09	13477
0028> 0027	57.69	1.07	13477
0007> 0008	56.98	1.05	13477
0030> 0031	54.74	1.01	13477
	54.44	1.01	13477
[filtered out		1.01	10111
	======================================		
1	5403.22		
AVG	4.67	0.09	1224



Intel[®] VTuneTM Profiler

Analyze & Tune Application Performance

Intel[®] VTune™ Profiler

Ana	alysis Configuration Collecti	on Log	Summary	Bottom-up	Calle	er/Callee	Top-down 1	Free Plat	form		
rou	ping: Function / Call Stack									~ % [2]	2,0
				CPU Time	e 🔻		«	Context S	witch Time 📧	Context Swit	c
	Function / Call Stack		ective Time by Poor 📒 Ok		» Over	Spin Time	Overhead Time	Wait Time	Inactive Time	Preemption	
up	dateBusinessAccount	7.915s				0s	0s	0s	0.055s	934	1
Ŧ	main\$omp\$parallel_for@269	7.915s				0s	0s	0s	0.055s	934	
	kmp_invoke_microtas	7.915s				0s	0s	0s	0.042s	815	
1	updateBusinessAccount	0s				0s	0s	0s	0.013s	119	
up	dateCustomerAccount	7.766s				0s	0s	0s	0.052s	1,111	
_	kmpc_atomic_fixed8_add	2.772s				0s	0s				
	kmpc_critical	Os				2.021s	0s	0s	0.014s	262	•
	>	<								>	L
): 🕇 🗕 🖝 🖻	5s	5.2s		5.6s	5.8s	6s 6.1		✓ Thread	~	1
	OMP Worker Thread #2 (TI	1.1.1.4.1			ر ار منا			<u>^</u>	Runr	ning	
	OMP Worker Thread #3 (TI				la se sut		ية بد البيلية بد		Contex	t Switches	
-		ار <u>سار کین از ر</u>	لمغاه كتفيل تلبينا	hi i di di seconda di dalam di dalam di seconda di seconda di seconda di seconda di seconda di seconda di secon	د الزار، عاملاً: 	tion of the links on the	با البير بالخطة	A fite of	Pre	emption	
-	rtmtest_openmp (TID: 12732)	ېسىل يېچەن	أرالتم أنرغت هين	متبيدية فأطبع	. بېلىرىغ خ	الابائل تهلسبا إحاله	زيد ينظرنه كروبات	ملته. به	Syr	nchronization	
	OMP Worker Thread #1 (TI	بأقتبة سلقت	يتأر فراجيه كر	هارتصير والطرن	ي ياسله	فت ورو وطرق في ا	الشيولين الررق	· م م م	CPU 🗠 CPU	Time	
	CPU Time		القادية أيتعارض						🖂 🖦 Spin	and Overh	
		<						>	🗌 🗖 🛡 CPU	CLK UNH	

https://software.intel.com/content/www/us/en/develop/tools/vtu ______ne-profiler/get-started.html

- Accurately profile C, C++, Fortran*, Python*, Go*, Java*, or any mix
- Optimize CPU, threading, memory, cache, storage & more
- Take advantage of <u>Priority Support</u>
 - Connects customers to Intel engineers for confidential inquiries (paid versions)
- A more accessible user interface provides a simplified profiling workflow
- Smarter, faster Application Performance Snapshot: Analyze CPU utilization of physical cores, pause/resume, more... (Linux*)

Start a new Project

🞏 Intel VTune Profiler		- 🗆 X	
Project Navigator	E E III ▷ L III O Welcome × Configure Analysis ×		
ttt	💯 Configure Analysis 👔	INTEL VTUNE PROFILER	 Use GUI
xtest	Local Host -	HOW Performance Snapshot -	• Or
	Launch Application -	Performance Snapshot ALGORITHM MICROARCHITECTURE	Command -Line
	Specify and configure your analysis target: an application or a script to execute. Application: C:\Users\hbockhor\Documents\\lssues\2020\OSC_04816644\test_intel\source_cor	Observation Image: Construction Hotspots Anomaly Detection Microarchitecture (preview) Exploration	Line
	Application parameters:	PARALLELISM I/O	
	Advanced	Threading HPC Performance Characterization	Get Command-Line
		ACCELERATORS PLATFORM AMALYSES	
		GPU Offload (preview) GPU (preview) Compute/Media Hotspots (preview)	



INTEL[®] ADVISOR

Intel[®] Advisor – Vectorization Advisor

Get breakthrough vectorization performance

- Faster Vectorization Optimization:
 - Vectorize where it will pay off most
 - Quickly ID what is blocking vectorization
 - Tips for effective vectorization
 - Safely force compiler vectorization
 - Optimize memory stride

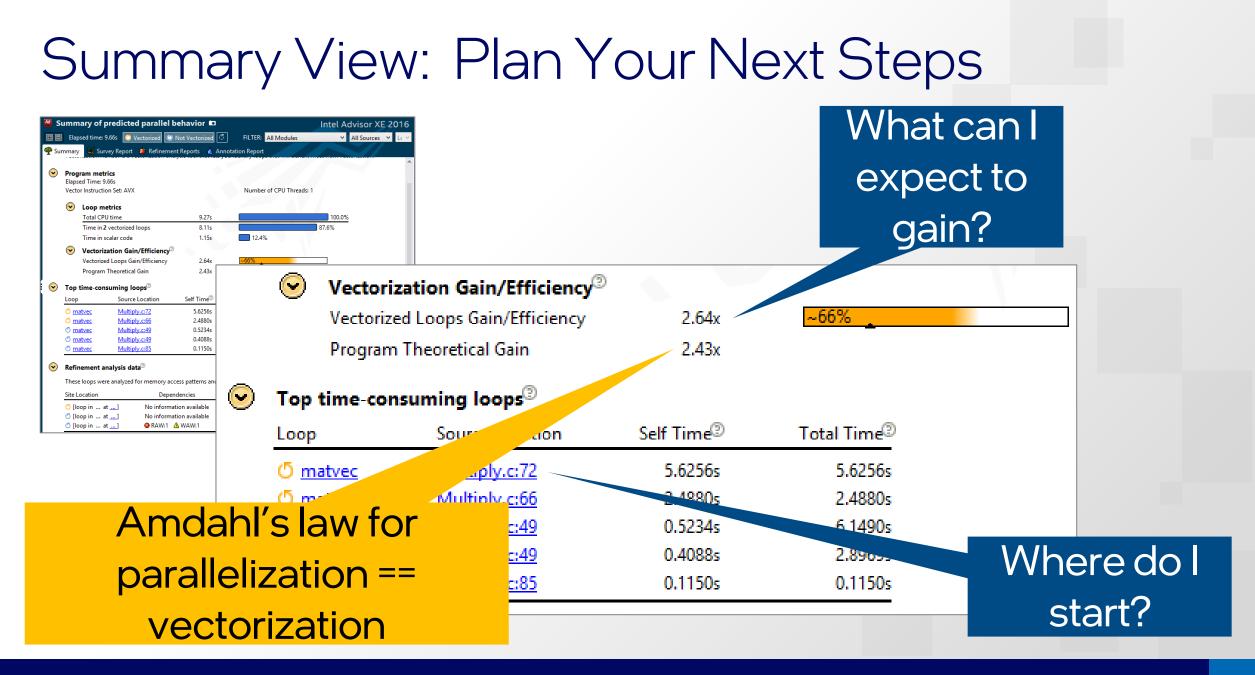
- The data and guidance you need:
 - Compiler diagnostics + Performance Data + SIMD efficiency
 - Detect problems & recommend fixes
 - Loop-Carried Dependency Analysis
 - Memory Access Patterns Analysis

FILTE		Loo	ps And Functio						٥	FF	Sma	rt Mode	° El Ai	DVISOR 2	م 018	Optimize for AVX-512
ROOFLINE	+ - Function Call Sites and Loops	۵	Vector Issues	Self Time 🕶	Total Time	Туре	FLOPS GFLOPS	AI	Why No Vectorization?		ed Loops Efficiency	Gain	1	- mp	>	with/without
	± ^[5] [loop in S252 at loops90.f:1172]	•	💡 1 Possible	3.129s 7.0%	3.129	Vectorized	0.191	0.115	🖬 1 vectorizat	AVX2	17%	1.36x	4; 8	99; 6; 1; 1		accessto
	☑ ⁽⁵] [loop in S2101 at loops90.f:1749]	✓	2 Possible	2.765s 6.2%	2.765s	Scalar	0.1421	0.067	vectorizatio					12		AVX-512
	⊕ ^[] [loop in s442_\$omp\$parallel_for		💡 1 Ineffecti	. 1.492s 3.4%	1.492s	Vectorized+	0.5861	0.165		AVX2	14%	1.09x	8	30; 1; 3		
				1.108s 2.5%	1.108s	Vector Funct	3.9111	0.156		AVX2						hardware
	⊕ ^[] [loop in S353 at loops90.f:2381]		9 1 Possible	0.989s 2.2%	0.989s	Vectorized (2.0231	0.134		AVX2	27%	2.16x	8	6; 4; 1	\sim	
	< >	<													>	

Part of oneAPI Base Toolkit

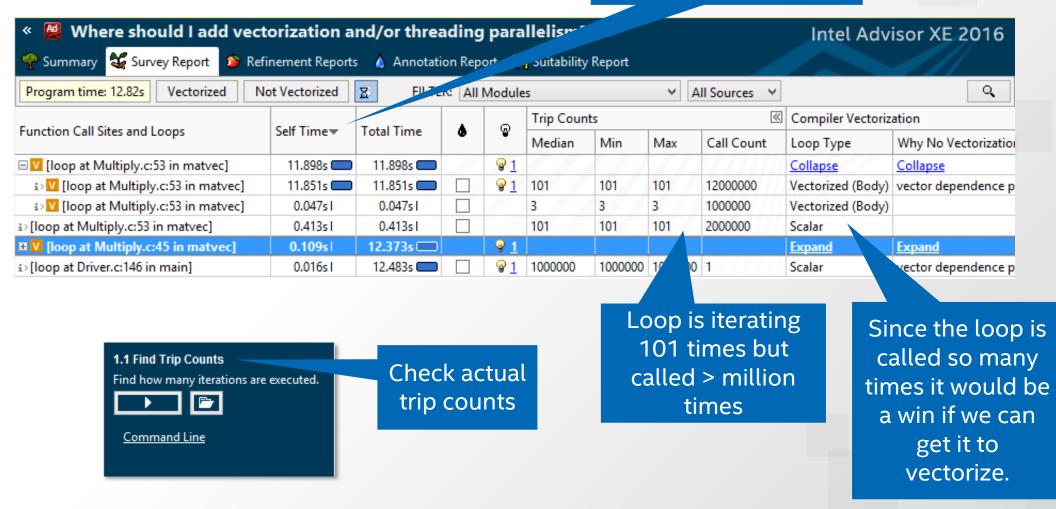
software.intel.com/advisor

intel



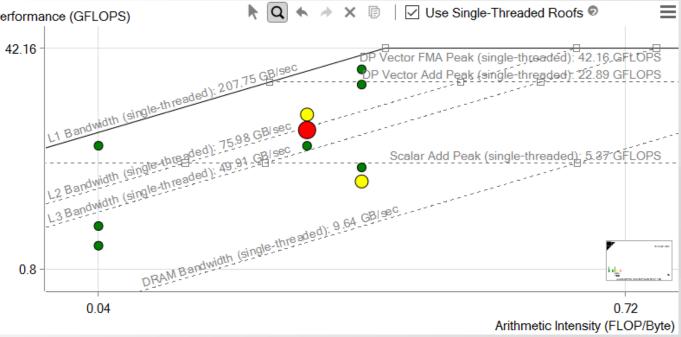
Critical Data Made Easy Loop Trip Counts

Knowing the time spent in a loop is not enough!



What is a Roofline Chart?

- A Roofline Chart plots application performance against hardware limitations.
 Performance (GFLOPS)
 Q * X © Use Single-Threaded Roofs
 - Where are the bottlenecks?
 - How much performance is being left on the table?
 - Which bottlenecks can be addressed, and which should be addressed?
 - What's the most likely cause?
 - What are the next steps?



Roofline first proposed by University of California at Berkeley: <u>Roofline: An Insightful Visual Performance Model for Multicore Architectures</u>, 2009 Cache-aware variant proposed by University of Lisbon: <u>Cache-Aware Roofline Model: Upgrading the Loft</u>, 2013

Advisor Resources

Intel[®] Advisor

- Product page overview, features, FAQs...
- What's New?
- Training materials <u>Cookbooks</u>, <u>User Guide</u>, Tutorials
- Support Forum
- Online Service Center Secure Priority Support

Additional Analysis Tools

- <u>Intel® VTune™ Profiler</u> performance profiler
- Intel® Inspector memory and thread checker/ debugger
- Intel® Trace Analyzer and Collector MPI Analyzer and Profiler

Additional Development Products

Intel[®] oneAPI Toolkits







ITAC for MPI Analysis

Efficiently Profile MPI Applications

Intel® Trace Analyzer & Collector

Helps Developers

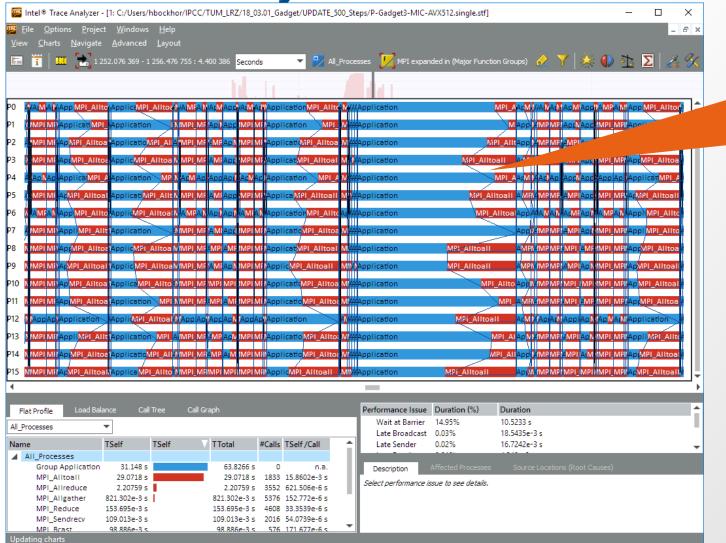
- Visualize & understand parallel application behavior
- Evaluate profiling statistics & load balancing
- Identify communication hotspots

Features

- Event-based approach
- Low overhead
- Excellent scalability
- Powerful aggregation & filtering functions
- Idealizer
- Scalable

<u>File O</u> ptions <u>P</u> ro View <u>C</u> harts <u>N</u> aviga	ject <u>W</u> indows <u>H</u> elp ite <u>A</u> dvanced <u>L</u> ayou	t					
👔 🔝 🛃 69. 17	73 734 - 69.372 292 : 0.198	8 558 Seconds 🔻	All_Nodes 🦻	Major Function	Groups 췵 🍸 🔆	• 🜗 🐴	<i>L</i> : 🛠
69.18 s	69.2 69.20 s	2 5 69.24 5	69.26 =	69.28 =	69.30 s 69.32 s	69.34 :	5 69.36 s
node1 Application	ApplidApApplication		ApplidArApplication		ApplidArApplication		
	< /				~ //		
node 1-mic0 MPI	AMPI Application	MPI	AMPI (Application	MPI	AMPI (Application	MPI	AMPI Applicatio
node1-mic1	AMPI. Application	MPI	AIMPI Application	MPI	AIMPI (Application	MPI	AIMPI Application
node2 Application							
node2 Application	ApplidApApplication		Appli: Application		ApplidArApplication		ApplidArApplicati
node2-mic0	AMPL Application	MPI	AMPI (Application	MPI	AMPI (Application	MPI	Art MPI Application
node2-mic1	ArMPI Application	MPI	AMPI Application	4MPI	ArMPI (Application	MPI	AIMPI AApplicatio
node3 Application	ApplidApApplication		NApplidApApplication		Appli:Application		ApplidApApplicatio
node3-mic0	AMPL Application	MPI	AMPI (Application	MPI	AMPI (Application	MPI	AMPI Application
node3-mic1.MPI	AIMPI. Application	MPI	AMPI Application	MPI	Ar MPI (Application	MPI	A MPI (Application
node4 Application	ApplicApApplication		ApplicApplication	<u> </u>	Appli: ArApplication		ApplicApplicatio
node4-mic0 AMPI	AMPL Application	MPI	Application	MPI	AN IPI (Application	MPI	AMIPI Application
node4-mic1//MPI	ArMPI Application	MPI	AIMPI (Application	MPI	AMPI (Application	MPI	AIMPI AApplication

ITAC Analysis



High Load imbalance causes MPI_Alltoall time

Online Resources

- Intel[®] MPI Library product page.
 - www.intel.com/go/mpi
- Intel[®] Trace Analyzer and Collector product page
 - www.intel.com/go/traceanalyzer
- Intel[®] Clusters and HPC Technology forums
 - <u>http://software.intel.com/en-us/forums/intel-clusters-and-hpc-technology</u>

Intel Modules installed on Juwels

Compiler:

Advisor:

- VTune + APS:
- check available: default:
- check available: default:
- check available default:
- Intel MPI: check available: default:
- Intel MKL: check available: default:

\$ module spider Intel \$ module load Intel \$ module spider vtune \$ module load VTune \$ module spider advisor \$ module load Advisor

\$ module spider intelMPI \$ module load IntelMPI

\$ module spider mkl \$ module load imkl

How to start?

- Compile with minimal options and run with APS (will provide tuning tips)
- Compile with -xhost and and check timing and APS report
- Optional! Compile with –xhost and –no-vec disables vectorization. Compare with previous timing
- Use: VTune Profiler: \$ module load VTune/<version>
- Use: Advisor: \$ module load Advisor/<version>
- Google for Intel related topics \rightarrow Intel Developer Zone etc.
- For APS/VTune add to your batch job: #SBATCH --disable-perfparanoid
- Please set thread affinity e.g.: \$ export KMP_AFFINITY=scatter,verbose This can speed up OMP programs up to 10X!
- Any questions: Heinrich.Bockhorst@Intel.com

Notices & Disclaimers

Intel technologies may require enabled hardware, software or service activation. Learn more at intel.com or from the OEM or retailer.

Your costs and results may vary.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Optimization Notice: 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. Notice Revision #20110804. https://software.intel.com/en-us/articles/optimization-notice

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors.

Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. See backup for configuration details. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See configuration disclosure for details. No product or component can be absolutely secure.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

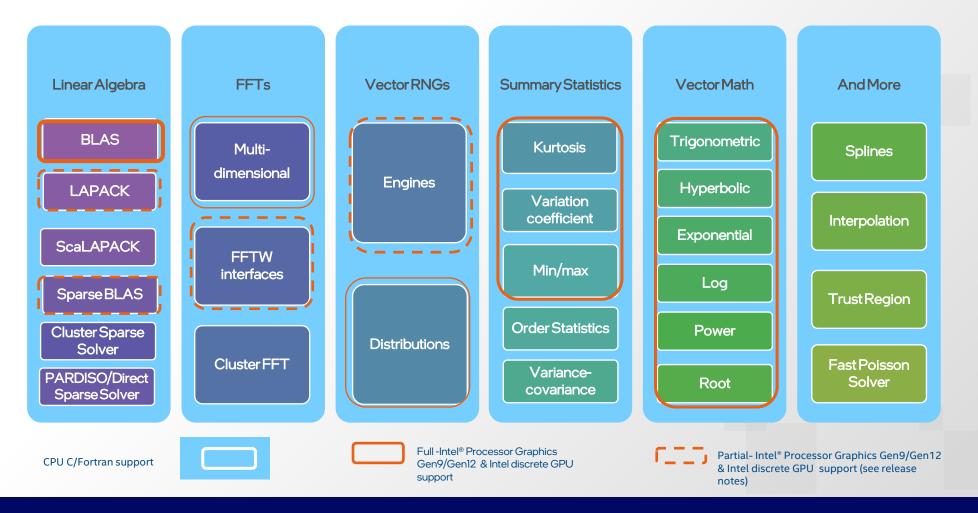
Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.



Intel[®] oneAPI Math Kernel Library (oneMKL)

What's Inside Intel[®] oneAPI Math Kernel Library (oneMKL)



intel.⁴⁵

What's New for Intel® oneAPI Math Kernel Library (oneMKL) 2021.2-2022.0

- Introduced GPU support for the following new functionality:
 - BLAS Batch & copy for unified shared memory(USM) & buffer APIs
 - Vector Statistics RNG multinomial, PoissonV, hypergeometric, negative binomial and binomial distributions.
 - BLAS Added SYCL support for in-place and out of place matrix copy/transposition
 - LAPACK Enabled C/Fortran OpenMP offload support for select functions.
 - Sparse BLAS Added support for variance matrix-matrix multiplication operations.
- General performance optimizations
- For detailed information please refer to the oneMKL <u>Release Notes</u>

Basic Vectorization Switches III

- Special switch in addition to CORE-AVX512: -qopt-zmm-usage=[keyword]
 - [keyword] = [high | low] ; Note: "low" is the default
 - Why choosing a defensive vectorization level?

Frequency drops in vectorized parts. Frequency does not immediately increase after the vectorized loop. Too many small vectorized loops will decrease the performance for the serial part.

Next steps

- Toolkits are free but maybe too large (> 10 GB). For this workshop you may download to your laptop: VTune, Advisor, Inspector
- Standalone tools download: <u>https://software.intel.com/content/www/us/en/develop/articles</u> /oneapi-standalone-components.html

#