Intel® Tuning for Juwels and Jureca

Dr. Heinrich Bockhorst

FZJ 2024, November 14th

intel®

Agenda

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

Cross-Architecture Programming for Accelerated Compute, Freedom of Choice for Hardware

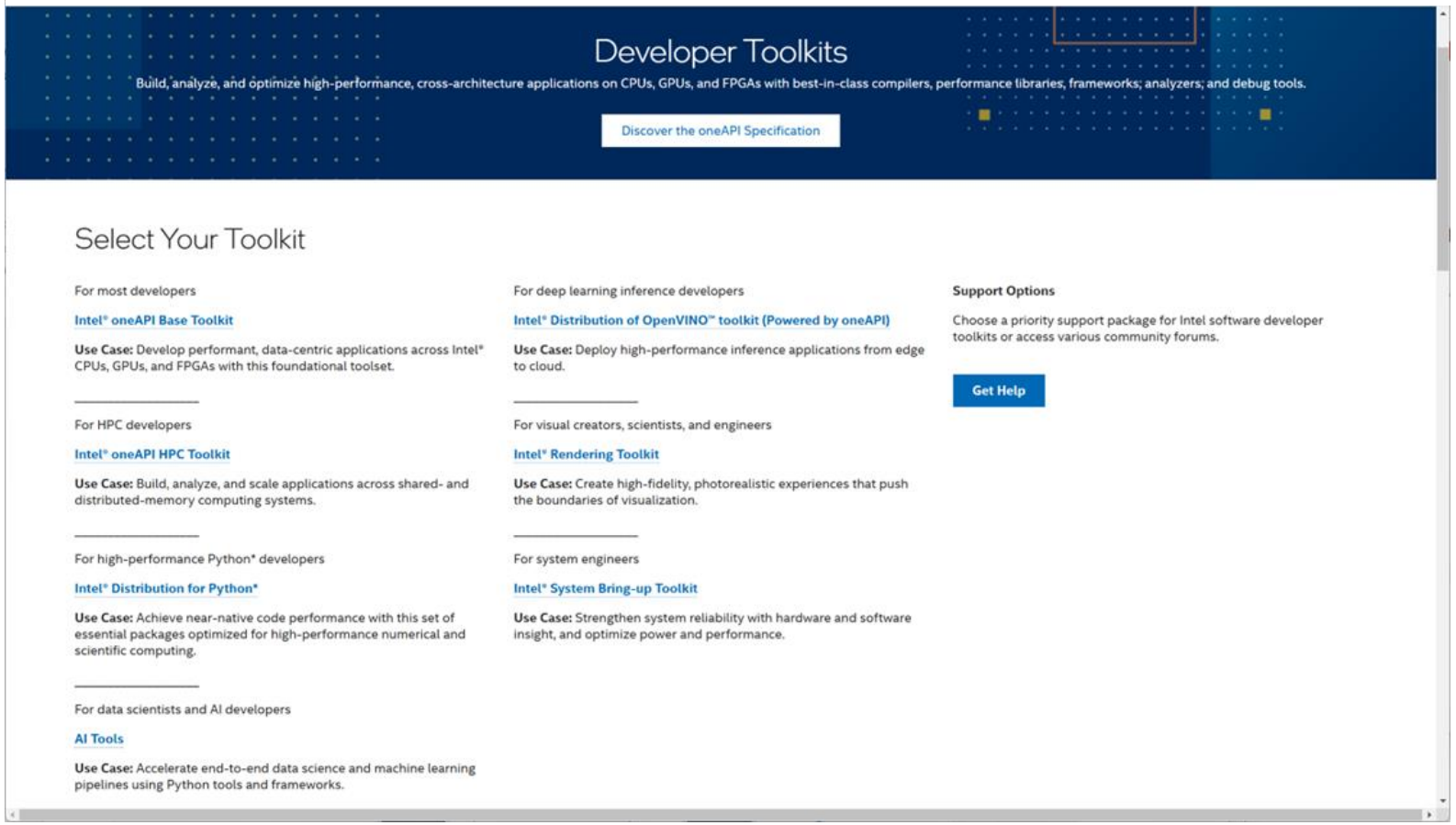
oneAPI: Industry Initiative & Intel Products

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

intel®

All information provided in this deck is subject to change without notice.
Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Toolkits for Developer: free download



Software for CPU and GPU

- Compiler (C++/SYCL) supports CPU and GPU from different Vendors
- GPU support not only for Intel GPU using plugins from codeplay:

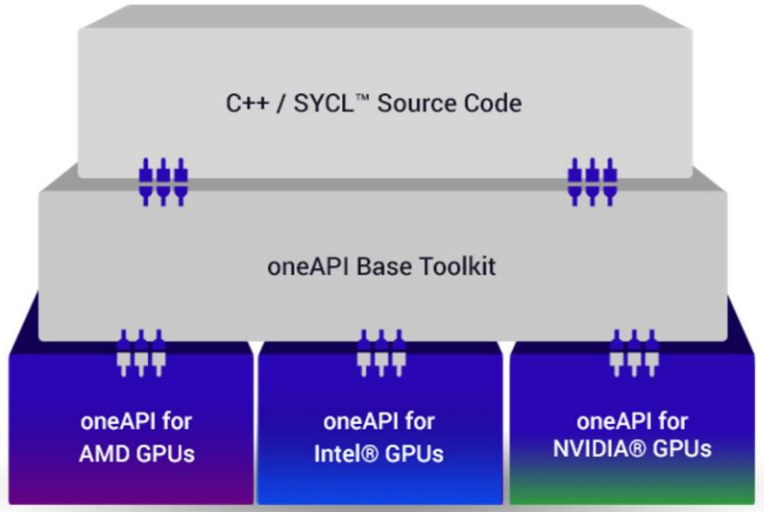
  https://developer.codeplay.com/products/oneapi/nvidia/2025.0.0/guides/index
- Single executable for Intel, NVIDIA and AMD possible
- FZJ may use plugin in upcoming stages
- Tools can be used for all devices

oneAPI for NVIDIA & AMD GPUs

- Free download of binary plugins to Intel® oneAPI DPC++/C++ Compiler:
  - Nvidia GPU
  - AMD GPU
- No need to build from source!
- Plug-ins updated quarterly in sync with SYCL 2020 conformance & performance

Priority Support

- Available through Intel, Codeplay & our channel
- Requires Intel Priority Support for Intel oneAPI DPC++/C++ Compiler
- Intel takes first call, Codeplay delivers backend support
- Codeplay provides access to older plug-in versions



Nvidia GPU plug-in

AMD GPU plug-in

Codeplay blog

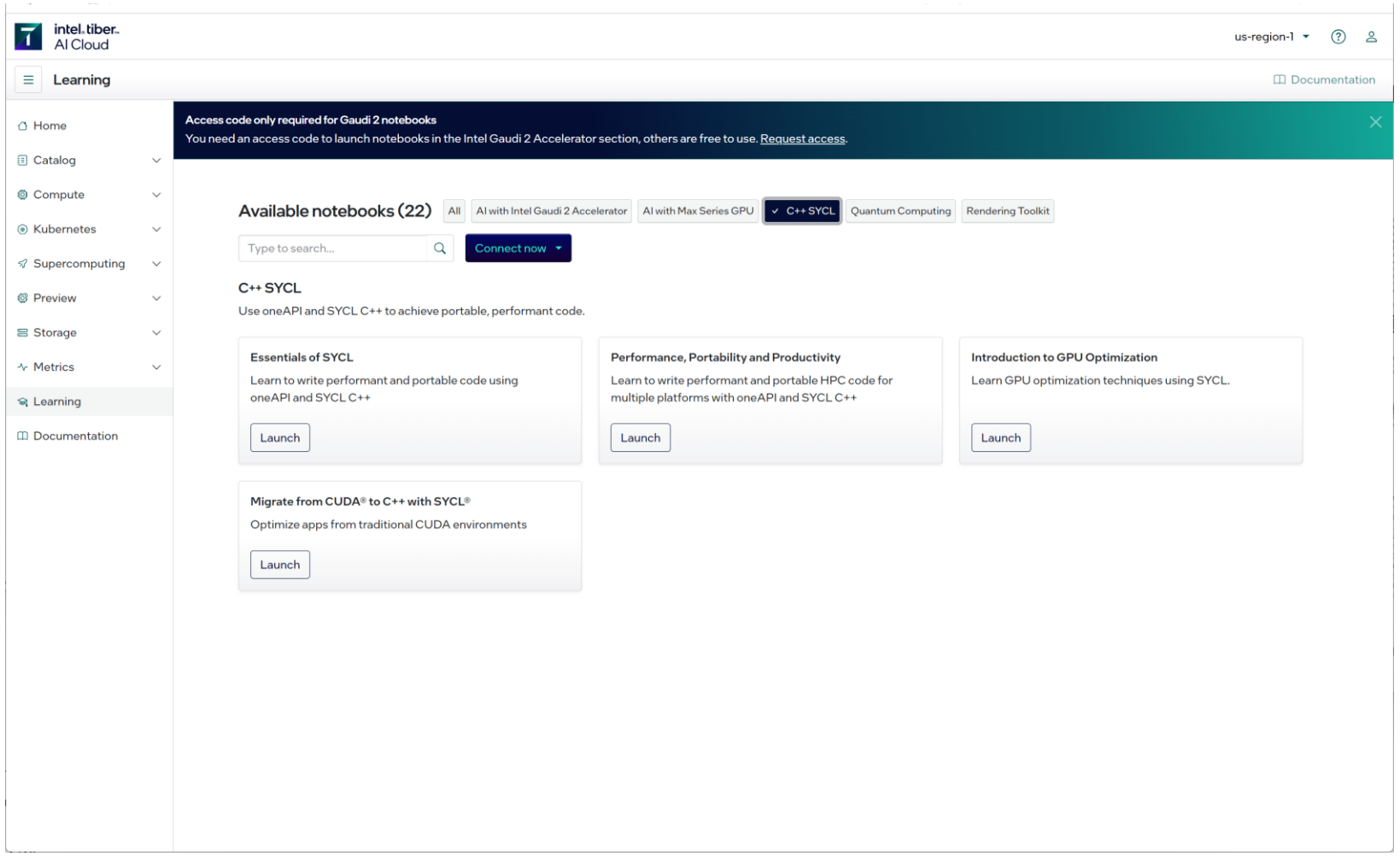
*Other names and brands may be claimed as the property of others. SYCL is a trademark of the Khronos Group Inc.

Single source for multiple devices and vendors

- On SC24 there will be a portability demo showing large production code running on different vendor cards.
  https://sc24.supercomputing.org/

- Check out for C++/SYCL tutorials on cloud.intel.com

Intel® Tiber™ Cloud – cloud.intel.com



Intel® Compilers

Intel® Compilers

Intel Compiler	Target	OpenMP Support	OpenMP Offload Support	Included in oneAPI Toolkit
Intel® C++ Compiler Classic, IL0 *icc/icpc/icl*	CPU	Yes	No	Removed 2025
Intel® Fortran Compiler Classic, IL0 *ifort*	CPU	Yes	No	Removed 2025
Intel® Fortran Compiler, LLVM *ifx*	CPU, GPU	Yes	Yes	HPC
Intel® oneAPI DPC++/C++ Compiler, LLVM *icx/icpx*	CPU GPU*	Yes	Yes	Base/HPC

Compiler Binary Compatible and Linkable!

tinyurl.com/oneapi-standalone-components

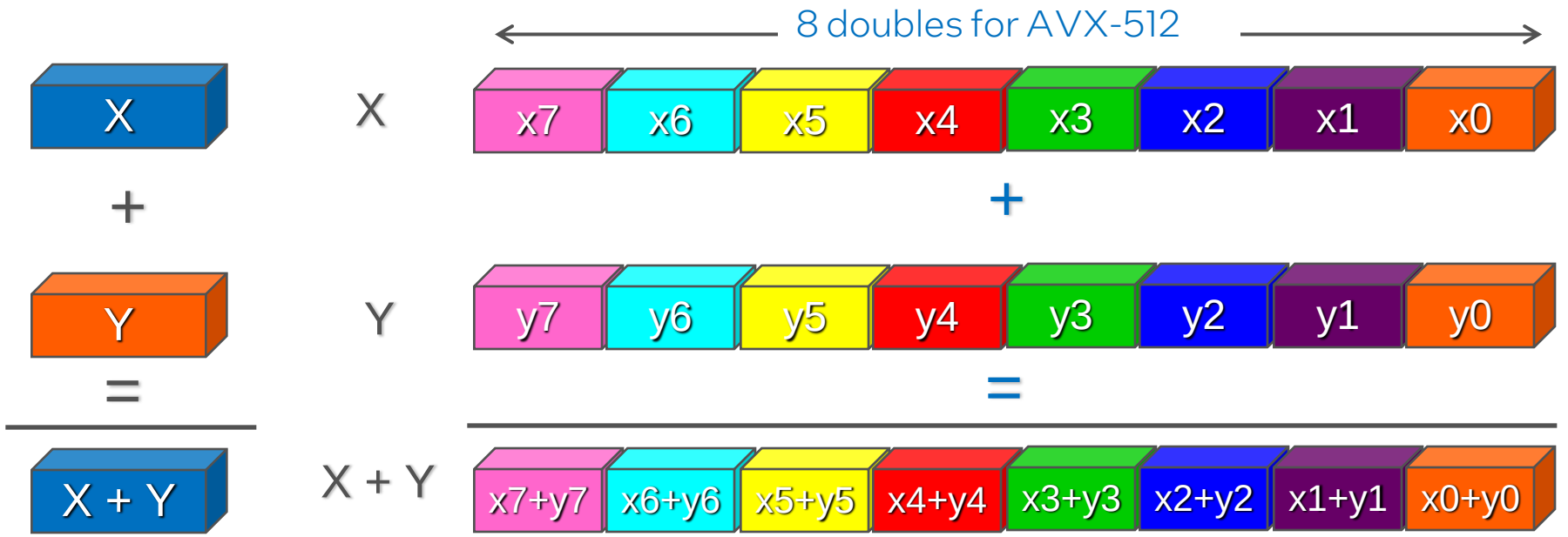
Common optimization options

	Linux* icx (icc)
Disable optimization	-O0
Optimize for speed (no code size increase)	-O1
Optimize for speed (default)	-O2
High-level loop optimization	-O3
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: Single Instruction, Multiple Data

```
for (i=0; i<n; i++) z[i] = x[i] + y[i];
```

- Scalar mode
  - one instruction produces one result
  - E.g. vaddss, (vaddsd)
- Vector (SIMD) mode
  - one instruction can produce multiple results
  - E.g. vaddps, (vaddpd)



Basic Vectorization Switches

- Special switch for icx, 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
- For AMD please check for the GNU flags e.g. **`-march=native`**

Please switch to icx/icpx Compiler!

- Deprecation of icc/ifort
- Check the user guide for supported flags:

  https://www.intel.com/content/www/us/en/docs/dpcpp-cpp-compiler/developer-guide-reference/2025-0/overview.htm

- Check results and compare with icc/icpc results:

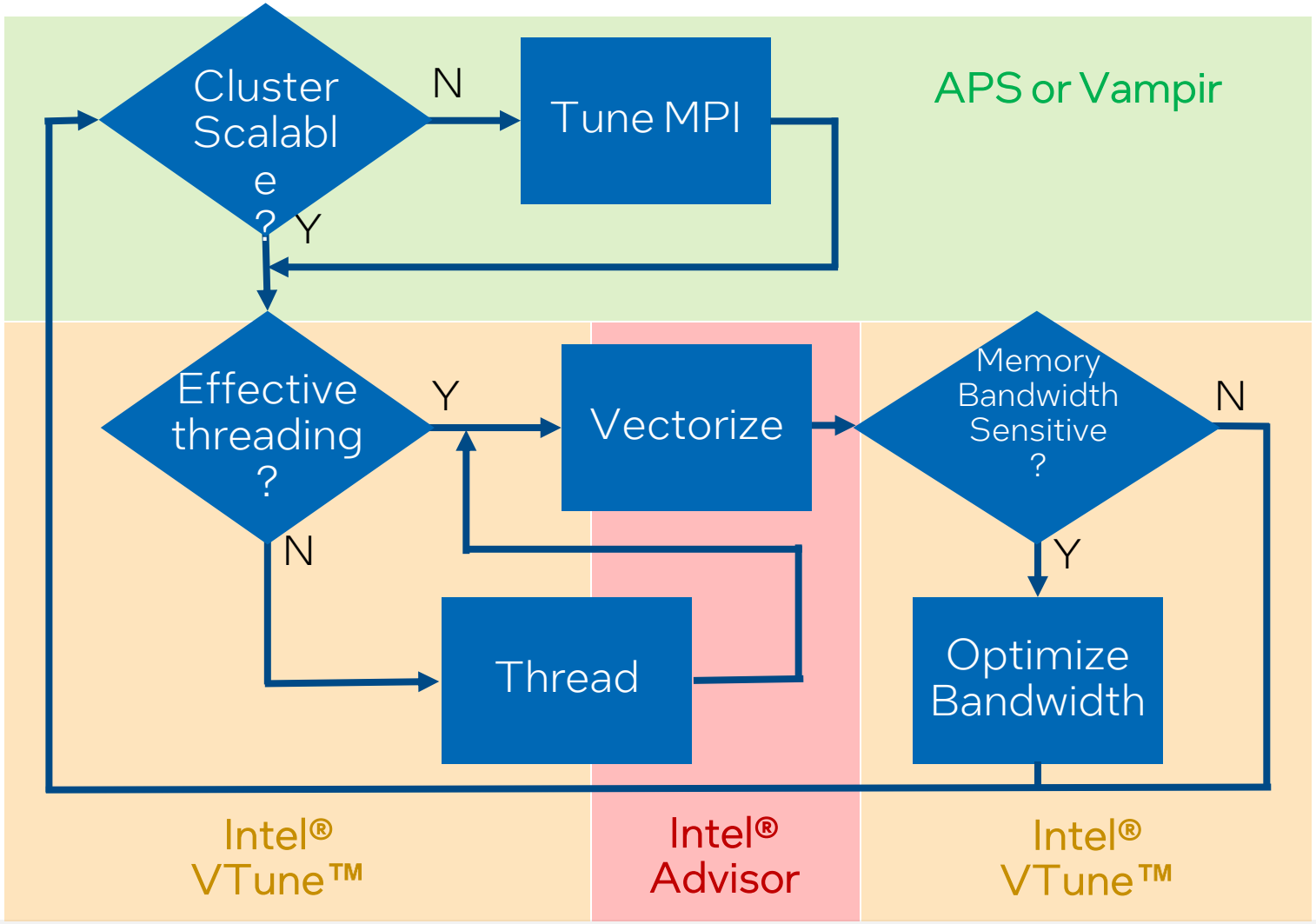
  -fp-model=fast is the default

  -fp-model=precise might help to reproduce previous results

- Good flags to start with: -O2 -xhost

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: **`$ module load VTune`**

APS Usage

Setup Environment

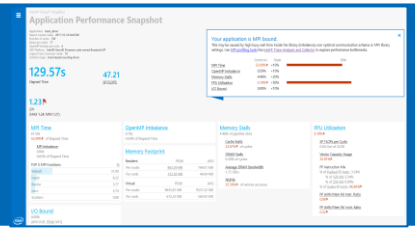
- $ source <path_to_vtune>/vtune_vars.sh # or load module

Run Application

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

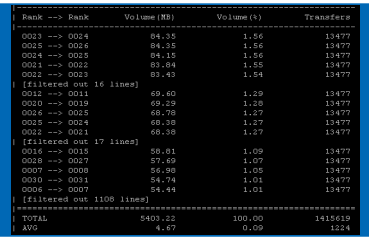
Generate Report on Result Folder

- $ aps –report <result folder>



Generate CL reports with detailed MPI statistics on Result Folder

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



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†

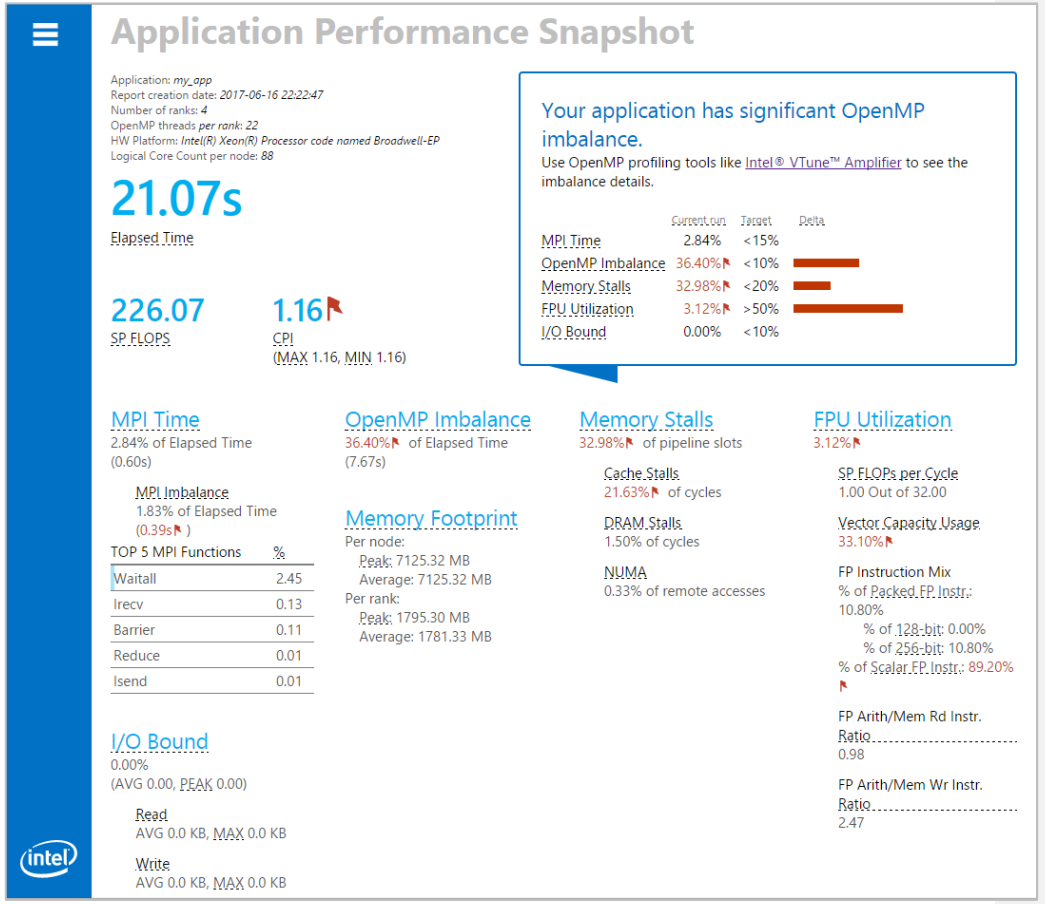
- Distributed MPI with or without threading
- Shared memory applications

Popular MPI Implementations Supported

- Intel® MPI Library
- MPICH & Cray MPI

Richer Metrics on Computation Efficiency

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



†MPI supported only on Linux*

APS Command Line Reports – Advanced MPI statistics

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

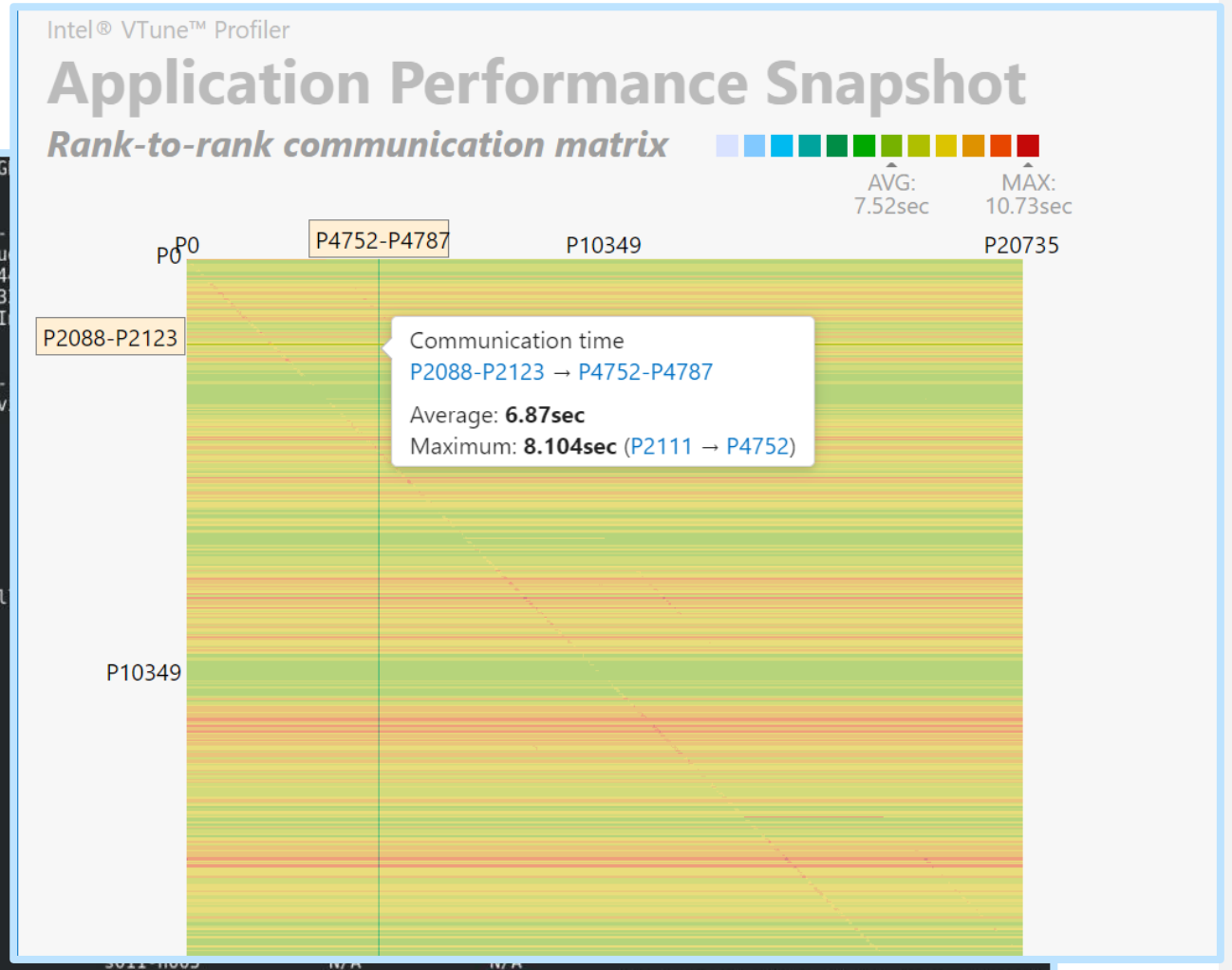
And many others – check
  - aps-report -help

Rank --> Rank	Volume(MB)	Volume(%)	Transfers
0023 --> 0024	84.35	1.56	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
0006 --> 0007	54.44	1.01	13477
[filtered out 1108 lines]			
TOTAL	5403.22	100.00	1415619
AVG	4.67	0.09	1224

Detailed reports

```
[root@nntpat98-144 aps_results]# aps --report -fDF
Loading 100.00%
| % - percentage of MPI functions total time
| Function summary for all Ranks
|-----------------------------------------------
|          Function             Time(sec)
|-----------------------------------------------
|             MPI_Waitall       972987.32
|                           Min: 0.0000
|                           Avg: 0.0001
|                           Max: 3.9528
|-----------------------------------------------
|             MPI_Allreduce     702927.29
|                           Min: 0.0000
|                           Avg: 0.0003
|                           Max: 1.8815
|-----------------------------------------------
|             MPI_Alltoallv     351070.07
|                           Min: 0.0000
|                           Avg: 0.0389
|                           Max: 17.3723
|-----------------------------------------------
|             MPI_Alltoall      178079.39
|                           Min: 0.0004
|                           Avg: 0.0030
|                           Max: 0.0197
|-----------------------------------------------
|             MPI_Barrier       105051.14
|                           Min: 0.0000
|                           Avg: 0.0650
|                           Max: 0.1494
|-----------------------------------------------
|             MPI_Isend         37178.85
|                           Min: 0.0000
|                           Avg: 0.0000
|                           Max: 0.2821
|-----------------------------------------------
|             MPI_Bcast         19726.74
|                           Min: 0.0000
|                           Avg: 0.0001
|                           Max: 1.4459
|-----------------------------------------------
|             MPI_Scatterv      8906.56
|                           Min: 0.0000
|                           Avg: 0.0015
|                           Max: 0.1564
|-----------------------------------------------
```

```
[root@nntpat98-144 aps_results]# aps --report --metrics="G
Loading 100.00%
| Metric Table
|-----------------------------------------------------------
Metric Name                        Node Name    Metric Valu
GPU Inbound PCIe Read, MB/s        s011-n004        207.14
GPU Inbound PCIe Read, MB/s        s011-n005        151.33
[root@nntpat98-144 aps_results]# aps --report --metrics="I
Loading 100.00%
| Metric Table
|-----------------------------------------------------------
Metric Name                                 Node Name    Dev
Inbound PCIe Read Per Device, MB/s          s011-n004
Inbound PCIe Read Per Device, MB/s          s011-n005
Inbound PCIe Read Per Device, MB/s          s011-n005
Inbound PCIe Read Per Device, MB/s          s011-n004
Inbound PCIe Read Per Device, MB/s          s011-n005
Inbound PCIe Read Per Device, MB/s          s011-n004
Inbound PCIe Read Per Device, MB/s          s011-n005
Inbound PCIe Read Per Device, MB/s          s011-n004
[root@nntpat98-144 aps_results]# aps --report --metrics=al
Loading 100.00%
GPU Time, s
GPU Time, s
GPU Time (% of Elapsed Time), % of Elapsed Time
GPU Time (% of Elapsed Time), % of Elapsed Time
GPU Utilization when Busy, %
GPU Utilization when Busy, %
GPU Occupancy, % of Peak Value
GPU Occupancy, % of Peak Value
GPU Inbound PCIe Read, MB/s
GPU Inbound PCIe Read, MB/s
GPU Inbound PCIe Write, MB/s
GPU Inbound PCIe Write, MB/s
GPU Outbound PCIe Read, MB/s
GPU Outbound PCIe Read, MB/s
GPU Outbound PCIe Write, MB/s
GPU Outbound PCIe Write, MB/s
Inbound PCIe Read Per Device, MB/s
Inbound PCIe Read Per Device, MB/s
Inbound PCIe Read Per Device, MB/s
```




```
s011-n004   N/A   GPU   Display controller: Intel Corporation De
s011-n005   N/A   GPU   Display controller: Intel Corporation De
s011-n005   N/A   GPU   Display controller: Intel Corporation De
```

APS potential issues

- If drivers are not available and for other hardware like AMD, you may use only mpi or/and openMP analysis (for Intel MPI):

```
$ aps --collection-mode=mpi[,openmp]
```

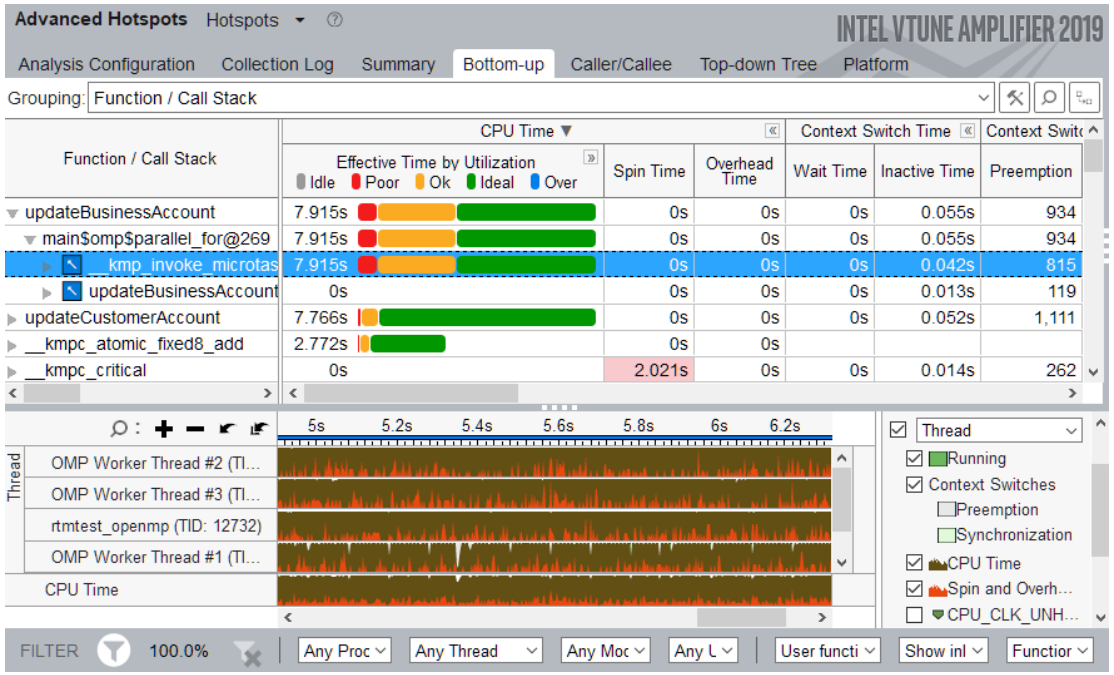
- If drivers are needed,you have to add this line to your batch job:

```
#SBATCH --disable-perfparanoid
```

Intel® VTune™ Profiler

Analyze & Tune Application Performance

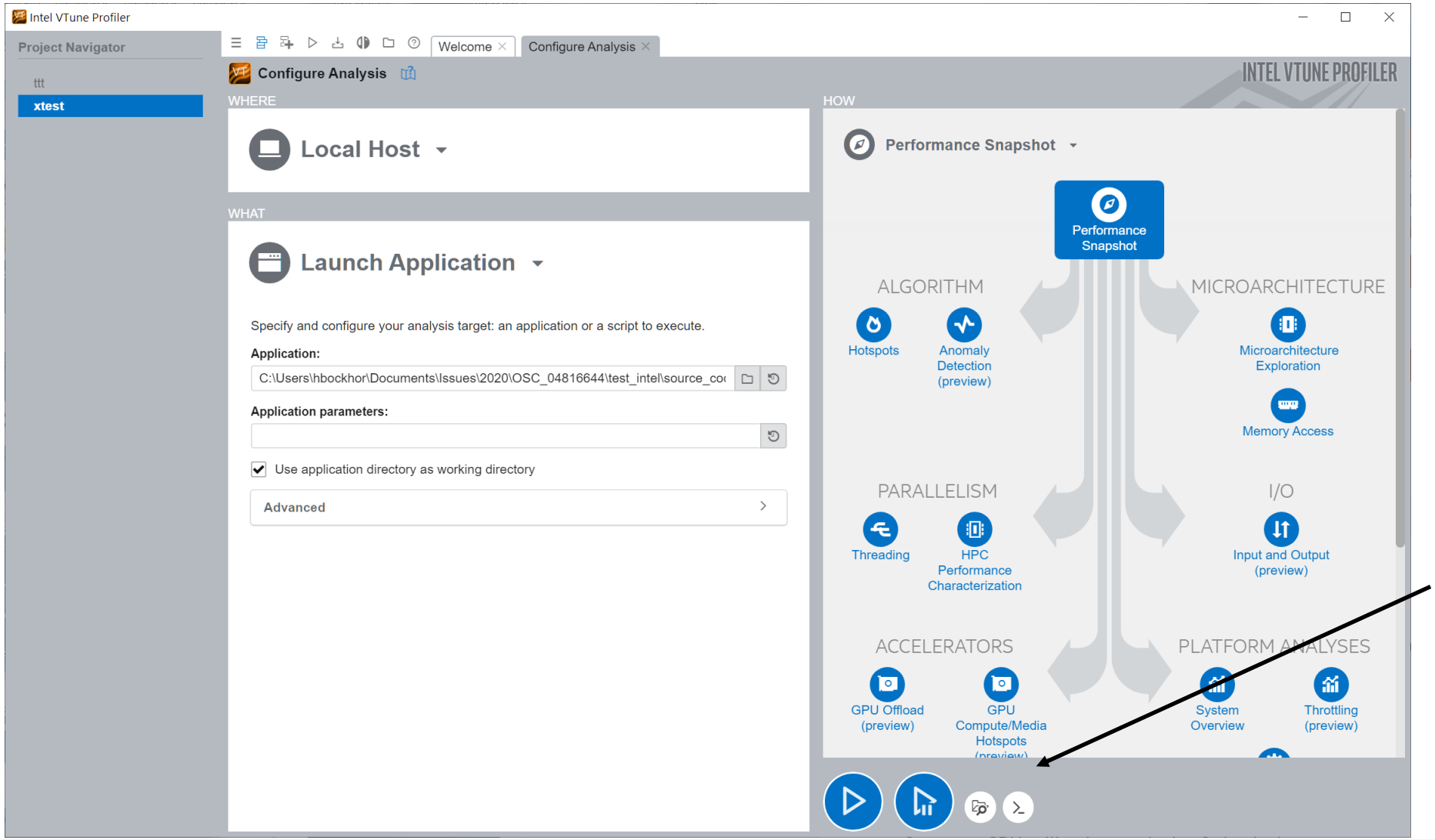
Intel® VTune™ Profiler



https://software.intel.com/content/www/us/en/develop/tools/vtune-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 Priority Support
  - 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



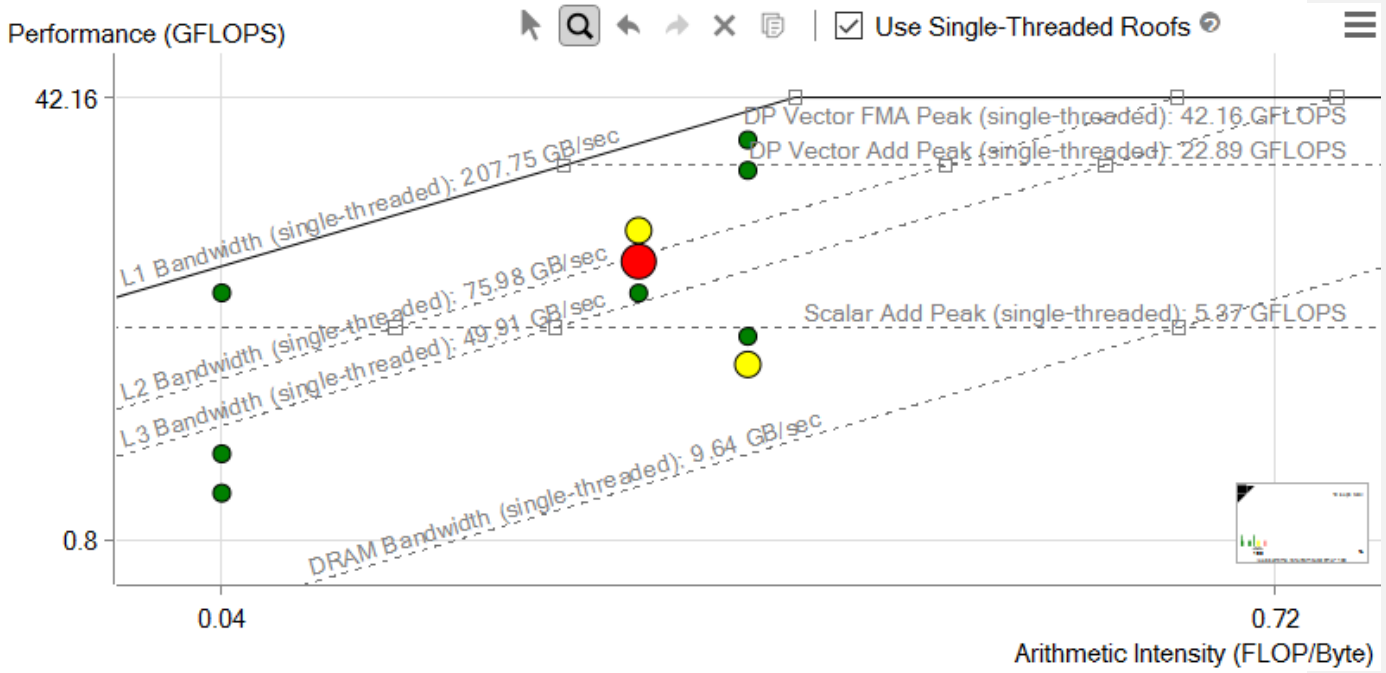
- Use GUI
- Or Command-Line

Get Command-Line

INTEL® ADVISOR

What is a Roofline Chart?

- A Roofline Chart plots application performance against hardware limitations.
  - 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:
Roofline: An Insightful Visual Performance Model for Multicore Architectures, 2009
Cache-aware variant proposed by University of Lisbon:
Cache-Aware Roofline Model: Upgrading the Loft, 2013

Advisor + VTune Resources

Intel® Advisor

- Product page – overview, features, FAQs…
- What's New?
- Training materials – Cookbooks, User Guide, Tutorials
- Support Forum
- Online Service Center - Secure Priority Support

Additional Analysis Tools

- Intel® VTune™ Profiler – performance profiler

Additional Development Products

- Intel® oneAPI Toolkits

INTEL® MKL
INTEL® MPI

Intel oneMKL (Math kernel libraries)

- https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl.html
- Math processing routines on CPU and GPU using C++, Fortran and python
- HPC and AI

Intel MPI

- https://www.intel.com/content/www/us/en/developer/tools/oneapi/mpi-library.html

- Is (Intel) GPU aware using device memory. Support for other GPUs is planed

Intel Modules installed on Juwels/Jureca

▪ Compiler:	check available:	$ module spider Intel
	default:	$ module load Intel
▪ VTune + APS:	check available:	$ module spider vtune
	default:	$ module load VTune
▪ Advisor:	check available	$ module spider advisor
	default:	$ module load Advisor
▪ Intel MPI:	check available:	$ module spider intelMPI
	default:	$ module load IntelMPI
▪ Intel MKL:	check available:	$ module spider mkl
	default:	$ module load imkl

How to start?

- Compile with minimal options and run with APS (will provide tuning tips)
- Compile with -O2 -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> and create a roofline report.
- Google for Intel related topics → 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

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**