ROOFLINE MODEL WITH INTEL® ADVISOR

Dr. Fabio Baruffa

Sr. HPC Apps. Engineer, Intel IAGS

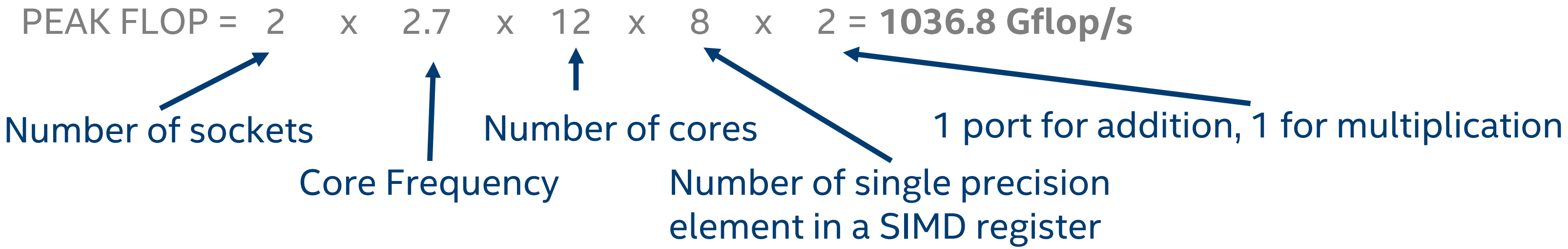
Platform PEAK FlopS

How many floating point operations per second

$$\text{Gflop/s} = \boldsymbol{min} \begin{cases} \boxed{\boldsymbol{Platform\ PEAK}} \\ \boldsymbol{Platform\ BW} \ * \boldsymbol{AI} \end{cases}$$

Theoretical value can be computed by specification

Example with 2 sockets Intel® Xeon® Processor E5-2697 v2



PEAK FLOP = 2 x 2.7 x 12 x 8 x 2 = **1036.8 Gflop/s**

Number of sockets

Core Frequency

Number of cores

Number of single precision element in a SIMD register

1 port for addition, 1 for multiplication

More realistic value can be obtained by running **Linpack**

=~ 930 Gflop/s on a 2 sockets Intel® Xeon® Processor E5-2697 v2

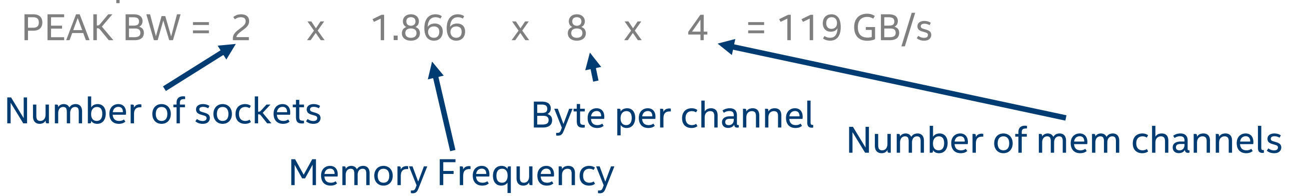
Platform PEAK bandwidth

How many bytes can be transferred per second

$$\text{Gflop/s} = \boldsymbol{min} \begin{cases} \boldsymbol{Platform\ PEAK} \\ \boxed{\boldsymbol{Platform\ BW}} * \boldsymbol{AI} \end{cases}$$

Theoretical value can be computed by specification

Example with 2 sockets Intel® Xeon® Processor E5-2697 v2



PEAK BW = 2 x 1.866 x 8 x 4 = 119 GB/s

More realistic value can be obtained by running **Stream**

=~ 100 GB/s on a 2 sockets Intel® Xeon® Processor E5-2697 v2

Drawing the Roofline

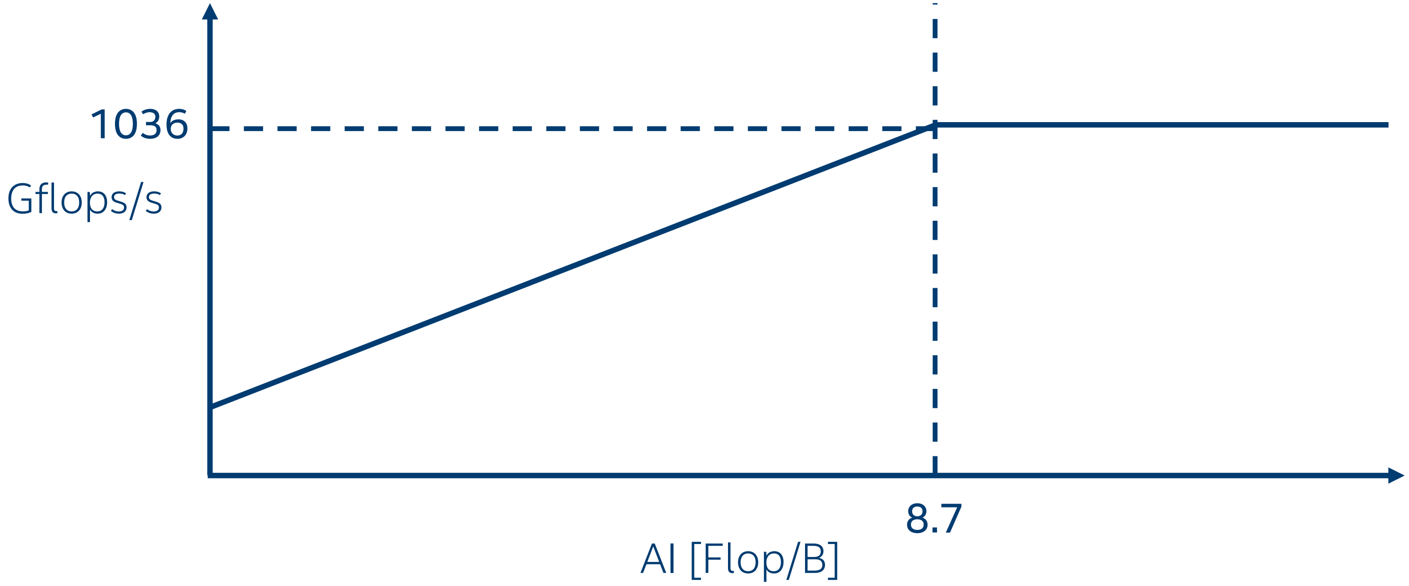
Defining the speed of light

$$\text{Gflop/s} = \boldsymbol{min} \begin{cases} \boldsymbol{Platform\ PEAK} \\ \boldsymbol{Platform\ BW} \ * \boldsymbol{AI} \end{cases}$$

2 sockets Intel® Xeon® Processor E5-2697 v2

Peak Flop = 1036 Gflop/s

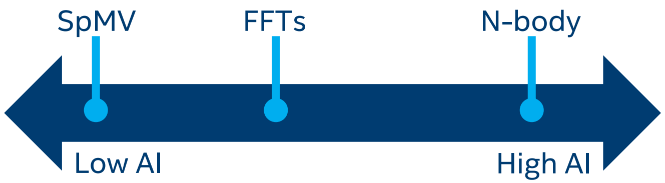
Peak BW = 119 GB/s



Plotting a Roofline Chart

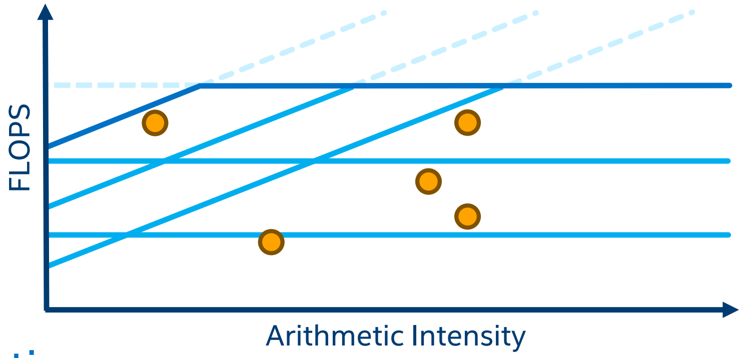
Roofline is based on FLOPS and Arithmetic Intensity (AI).

- **FLOPS**: **Fl**oating-Point **Op**erations / **S**econd
- **Arithmetic Intensity**: FLOP / Byte Accessed
  - Classic only counts DRAM traffic
  - Cache-Aware counts all memory



The lines are hardware limitations.

- Horizontal compute limitations
- Diagonal memory limitations

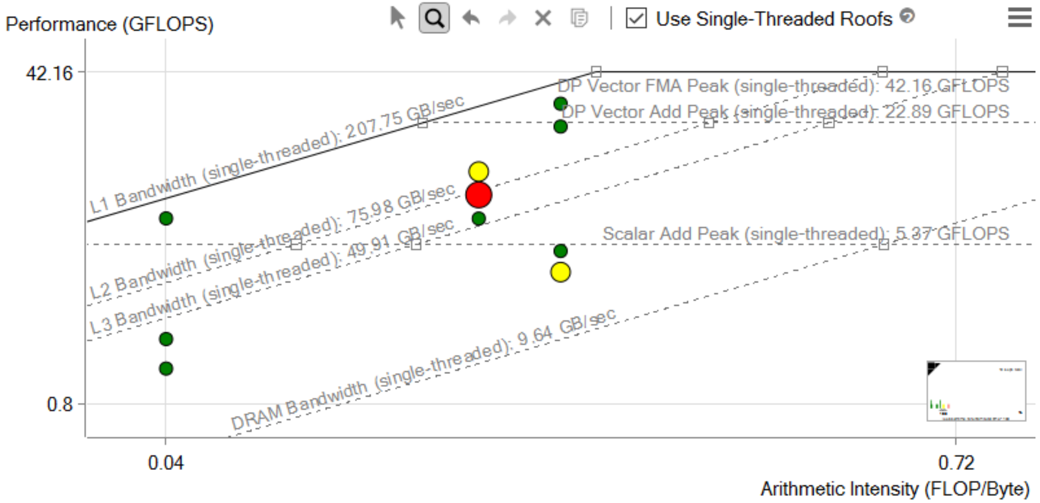


Dots represent loops/functions in the application.

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

Optimization Notice

Classic vs. Cache-Aware Roofline

Intel® Advisor uses the Cache-Aware Roofline model, which has a different definition of Arithmetic Intensity than the original (“Classic”) model.

Classical Roofline

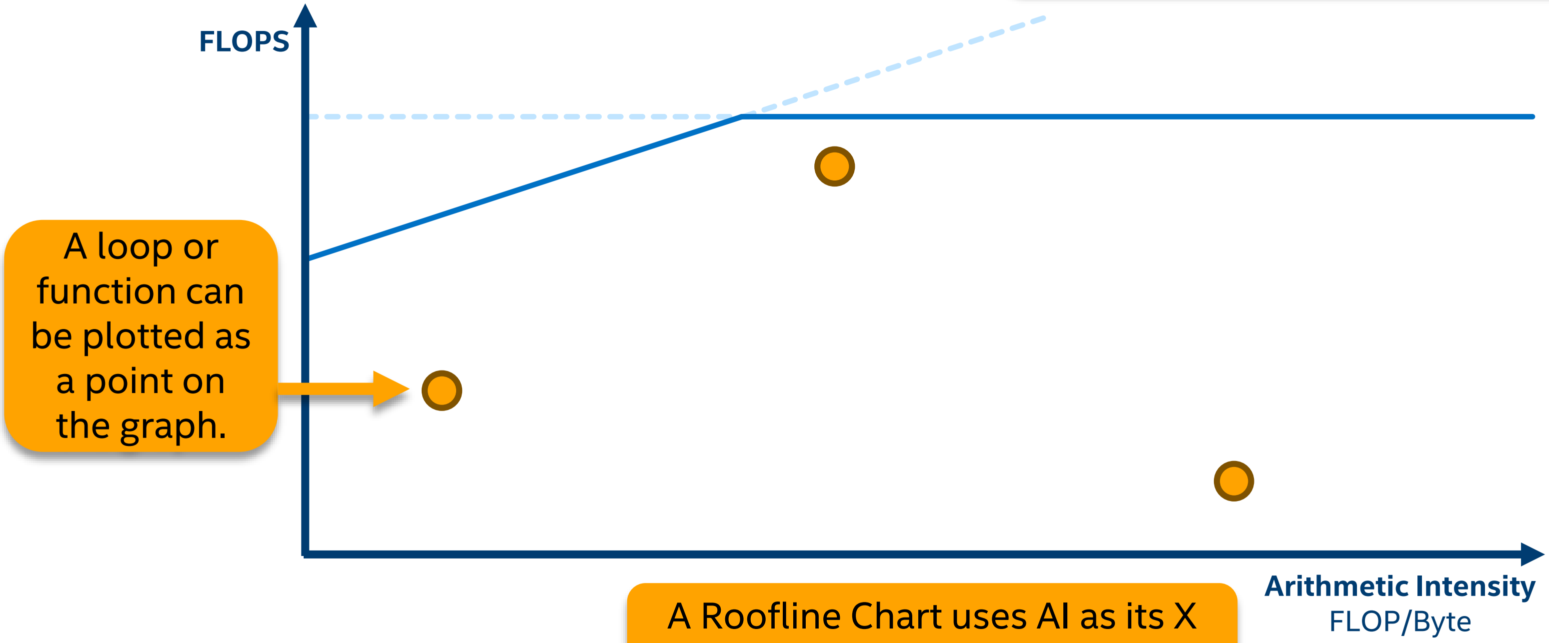
- Traffic measured from one level of memory (usually DRAM)
- AI may change with data set size
- AI changes as a result of memory optimizations

Cache-Aware Roofline

- Traffic measured from all levels of memory
- AI is tied to the algorithm and will not change with data set size
- Optimization does not change AI*, only the performance

**Compiler optimizations may modify the algorithm, which may change the AI.*

Plotting a Roofline Chart



The maximum FLOPS as a product of ops/byte (AI) and maximum bytes supplied per second is a diagonal line.

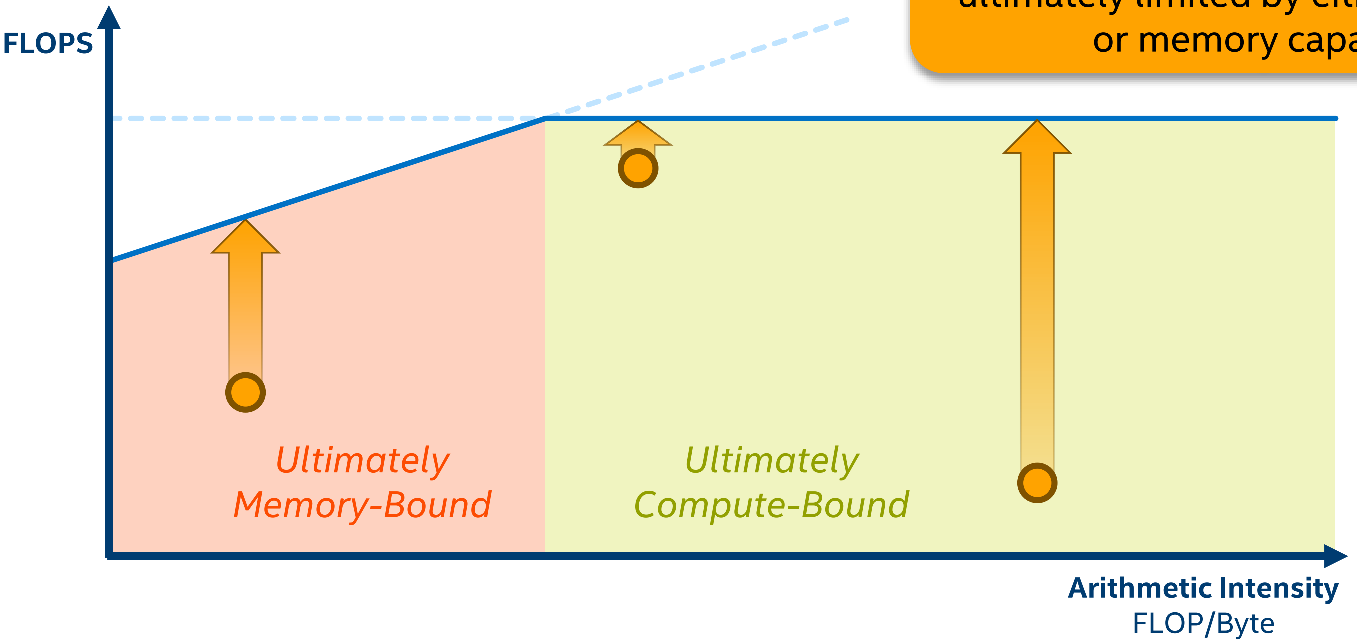
The CPU's maximum FLOPS can be plotted as a horizontal line.

A loop or function can be plotted as a point on the graph.

A Roofline Chart uses AI as its X axis and FLOPS as its Y axis.

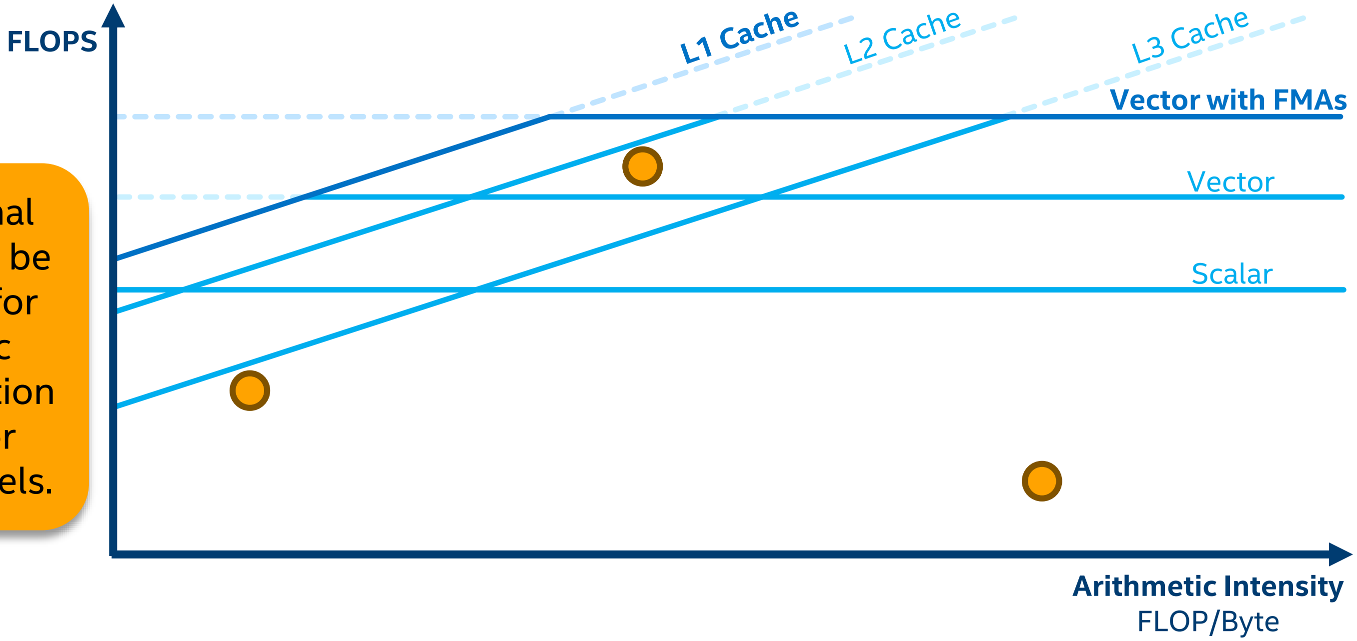
Ultimate Performance Limits

Performance cannot exceed the machine's capabilities, so each loop is ultimately limited by either compute or memory capacity.



Sub-Roofs and Current Limits

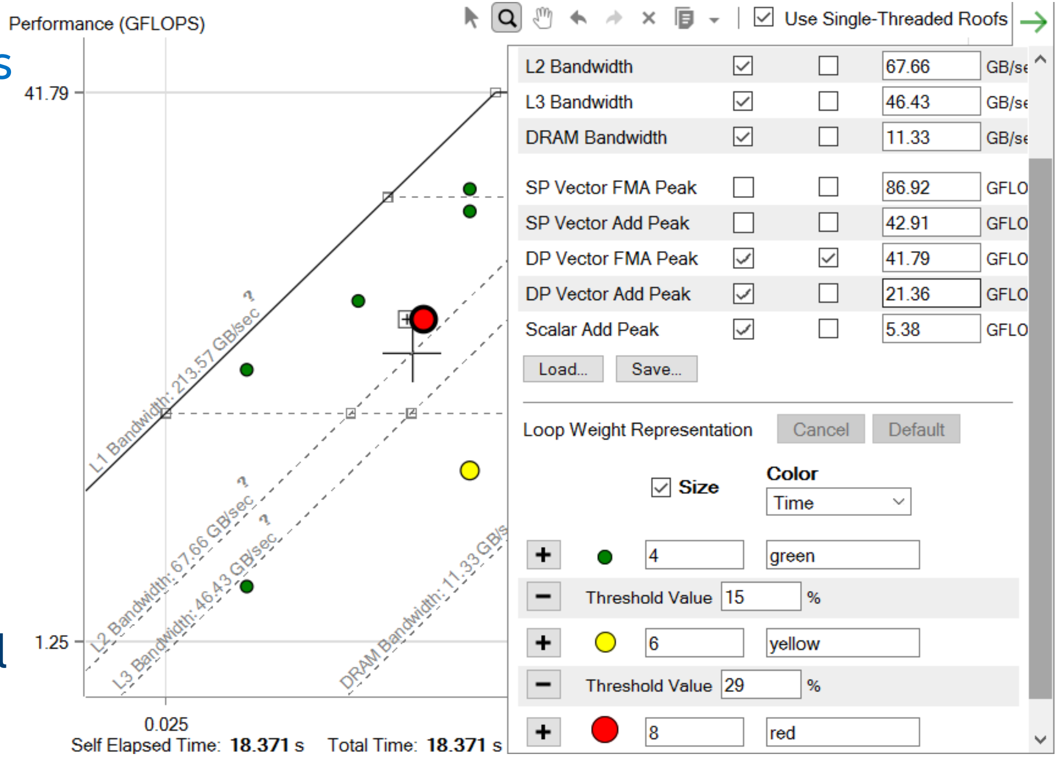
Additional roofs can be plotted for specific computation types or cache levels.



These sub-roofs can be used to help diagnose bottlenecks.

The Intel® Advisor Roofline Interface

- Roofs are based on benchmarks run before the application.
  - Roofs can be hidden, highlighted, or adjusted.
- Intel® Advisor has size- and color-coding for dots.
  - Color code by duration or vectorization status
  - Categories, cutoffs, and visual style can be modified.


Performance (GFLOPS)
41.79
1.25
0.025
L1 Bandwidth: 213.57 GB/sec
L2 Bandwidth: 67.66 GB/sec
L3 Bandwidth: 46.43 GB/sec
DRAM Bandwidth: 11.33 GB/s
Load...
Save...
Cancel
Default

☑ Use Single-Threaded Roofs

L2 Bandwidth	☑	□	67.66	GB/se
L3 Bandwidth	☑	□	46.43	GB/se
DRAM Bandwidth	☑	□	11.33	GB/se
SP Vector FMA Peak	□	□	86.92	GFLO
SP Vector Add Peak	□	□	42.91	GFLO
DP Vector FMA Peak	☑	☑	41.79	GFLO
DP Vector Add Peak	☑	□	21.36	GFLO
Scalar Add Peak	☑	□	5.38	GFLO

Loop Weight Representation

☑ Size

Color: Time

+	●	4	green
−	Threshold Value	15	%
+	●	6	yellow
−	Threshold Value	29	%
+	●	8	red

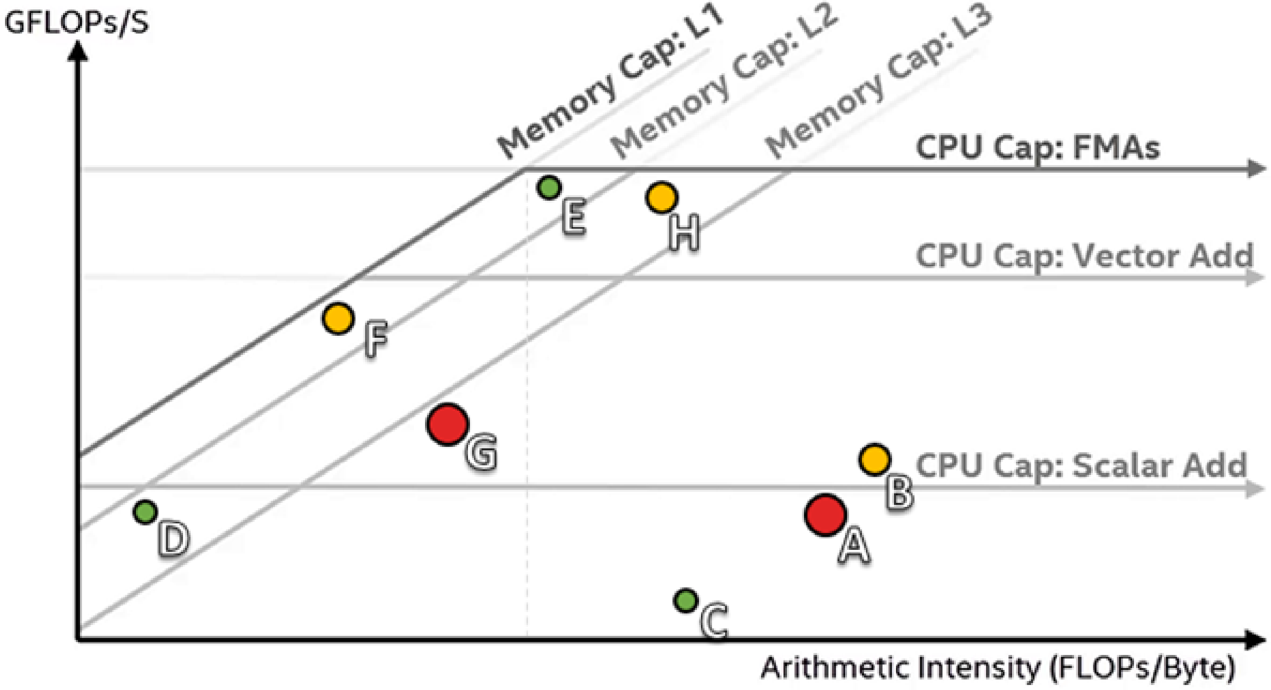
Self Elapsed Time: **18.371** s Total Time: **18.371** s

Optimization Notice

Identifying Good Optimization Candidates

Focus optimization effort where it makes the most difference.

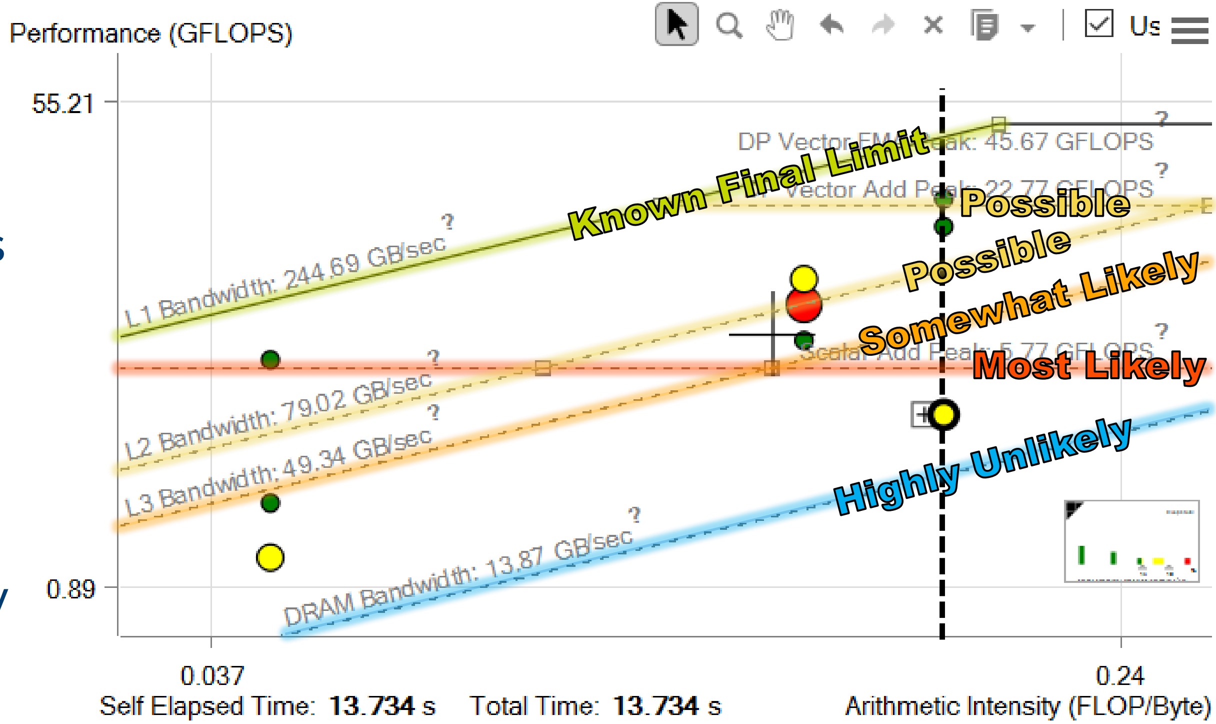
- Large, red loops have the most impact.
- Loops far from the upper roofs have more room to improve.



Identifying Potential Bottlenecks

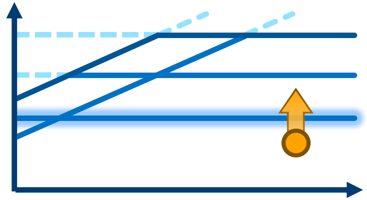
Final roofs *do* apply;
sub-roofs *may* apply.

- Roofs above indicate *potential* bottlenecks
- Closer roofs are the most likely suspects
- Roofs below may contribute but are generally not primary bottlenecks



Feature Synergy

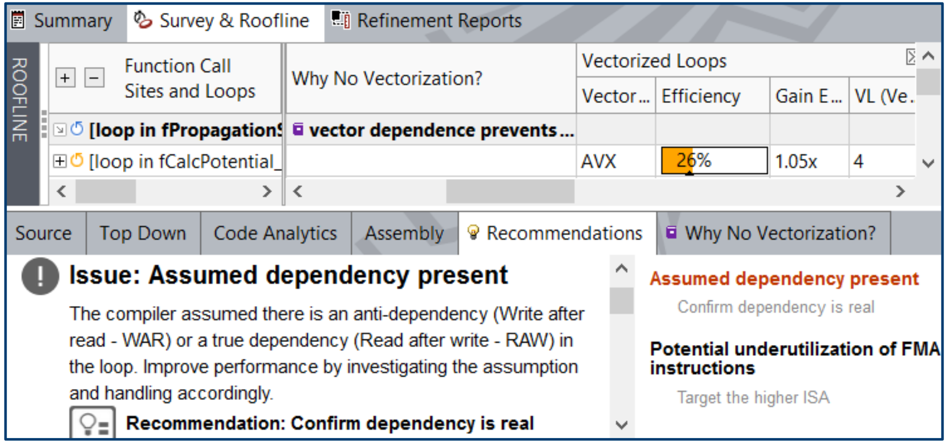
Overcoming the Scalar Add Peak



- Survey and Code Analytics tabs indicate vectorization status with colored icons.

  ↺ = Scalar ↺ = Vectorized

- "Why No Vectorization" tab and column in Survey explain what prevented vectorization.
- Recommendations tab may help you vectorize the loop.
- Dependencies determines if it's safe to force vectorization.



Function Call Sites and Loops	Why No Vectorization?	Vectorized Loops Vector...	Efficiency	Gain E...	VL (Ve...
[loop in fPropagation	**vector dependence prevents ...**				
[loop in fCalcPotential_		AVX	26%	1.05x	4

Issue: Assumed dependency present

The compiler assumed there is an anti-dependency (Write after read - WAR) or a true dependency (Read after write - RAW) in the loop. Improve performance by investigating the assumption and handling accordingly.

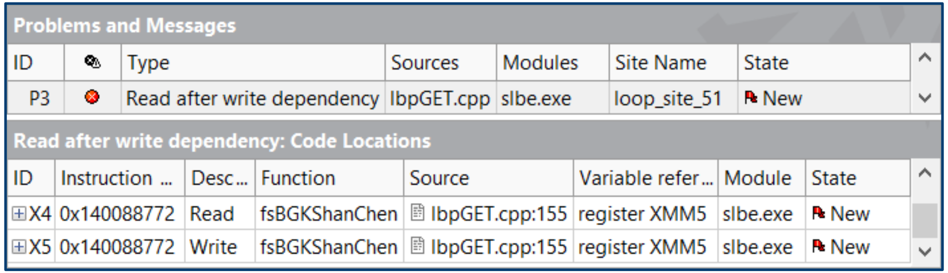
Recommendation: Confirm dependency is real

Assumed dependency present

Confirm dependency is real

Potential underutilization of FMA instructions

Target the higher ISA



Problems and Messages

ID		Type	Sources	Modules	Site Name	State
P3	●	Read after write dependency	lbpGET.cpp	slbe.exe	loop_site_51	New

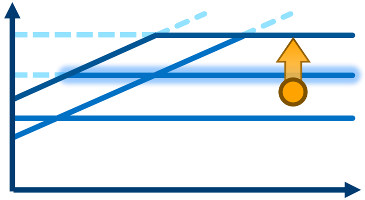
Read after write dependency: Code Locations

ID	Instruction ...	Desc...	Function	Source	Variable refer...	Module	State
X4	0x140088772	Read	fsBGKShanChen	lbpGET.cpp:155	register XMM5	slbe.exe	New
X5	0x140088772	Write	fsBGKShanChen	lbpGET.cpp:155	register XMM5	slbe.exe	New

Optimization Notice

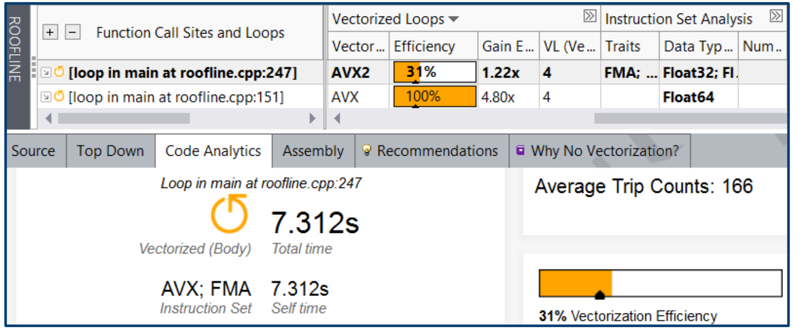
Feature Synergy

Overcoming the Vector Add Peak



Survey and Code Analytics display the vector efficiency and presence of FMAs.

- Recommendations may help improve efficiency or induce FMA usage.



Function Call Sites and Loops	Vector...	Efficiency	Gain E...	VL (Ve...	Traits	Data Typ...	Num...
[loop in main at roofline.cpp:247]	**AVX2**	**31%**	**1.22x**	**4**	**FMA; ...**	**Float32; Fl.**	
[loop in main at roofline.cpp:151]	AVX	100%	4.80x	4		**Float64**	

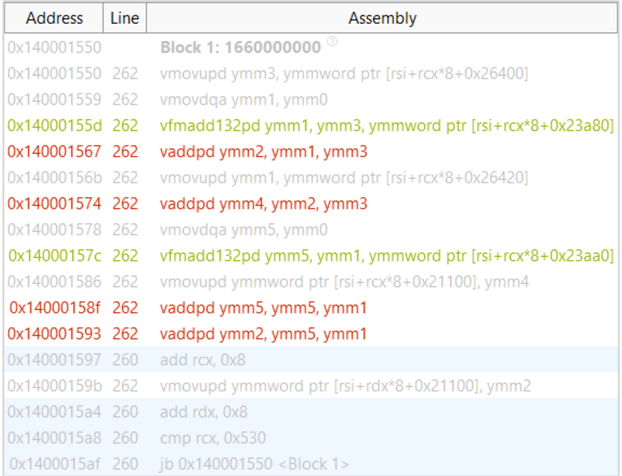
Loop in main at roofline.cpp:247

7.312s
Vectorized (Body) *Total time*

AVX; FMA 7.312s
Instruction Set *Self time*

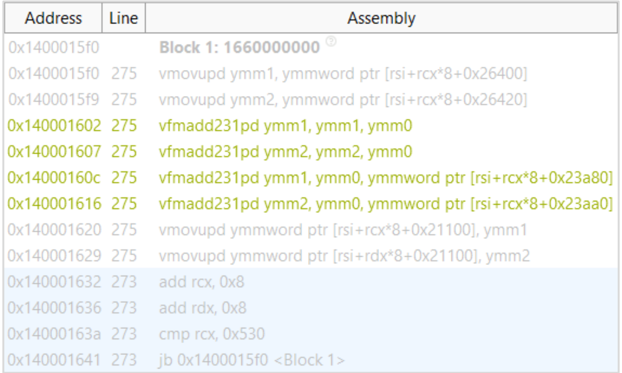
Average Trip Counts: 166

31% Vectorization Efficiency



Address	Line	Assembly
0x140001550		**Block 1: 1660000000**
0x140001550	262	vmovupd ymm3, ymmword ptr [rsi+rcx*8+0x26400]
0x140001559	262	vmovdqa ymm1, ymm0
0x14000155d	262	vfmadd132pd ymm1, ymm3, ymmword ptr [rsi+rcx*8+0x23a80]
0x140001567	262	vaddpd ymm2, ymm1, ymm3
0x14000156b	262	vmovupd ymm1, ymmword ptr [rsi+rcx*8+0x26420]
0x140001574	262	vaddpd ymm4, ymm2, ymm3
0x140001578	262	vmovdqa ymm5, ymm0
0x14000157c	262	vfmadd132pd ymm5, ymm1, ymmword ptr [rsi+rcx*8+0x23aa0]
0x140001586	262	vmovupd ymmword ptr [rsi+rcx*8+0x21100], ymm4
0x14000158f	262	vaddpd ymm5, ymm5, ymm1
0x140001593	262	vaddpd ymm2, ymm5, ymm1
0x140001597	260	add rcx, 0x8
0x14000159b	262	vmovupd ymmword ptr [rsi+rdx*8+0x21100], ymm2
0x1400015a4	260	add rdx, 0x8
0x1400015a8	260	cmp rcx, 0x530
0x1400015af	260	jb 0x140001550 <Block 1>

The Assembly tab* is useful for determining how well you are making use of FMAs.



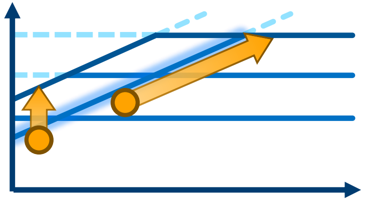
Address	Line	Assembly
0x1400015f0		**Block 1: 1660000000**
0x1400015f0	275	vmovupd ymm1, ymmword ptr [rsi+rcx*8+0x26400]
0x1400015f9	275	vmovupd ymm2, ymmword ptr [rsi+rcx*8+0x26420]
0x140001602	275	vfmadd231pd ymm1, ymm1, ymm0
0x140001607	275	vfmadd231pd ymm2, ymm2, ymm0
0x14000160c	275	vfmadd231pd ymm1, ymm0, ymmword ptr [rsi+rcx*8+0x23a80]
0x140001616	275	vfmadd231pd ymm2, ymm0, ymmword ptr [rsi+rcx*8+0x23aa0]
0x140001620	275	vmovupd ymmword ptr [rsi+rcx*8+0x21100], ymm1
0x140001629	275	vmovupd ymmword ptr [rsi+rdx*8+0x21100], ymm2
0x140001632	273	add rcx, 0x8
0x140001636	273	add rdx, 0x8
0x14000163a	273	cmp rcx, 0x530
0x140001641	273	jb 0x1400015f0 <Block 1>

**Color coding added for clarity.*

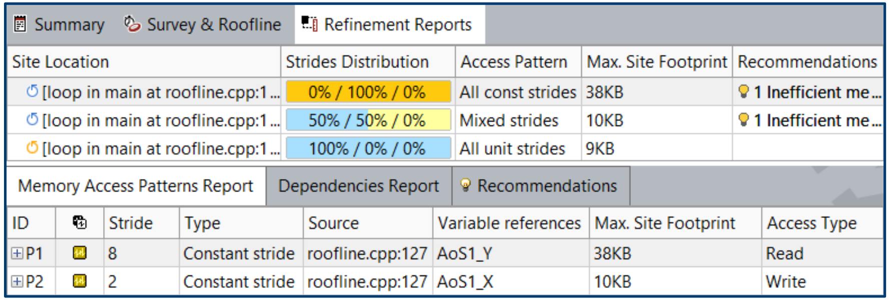
Optimization Notice

Feature Synergy

Overcoming the Memory Bandwidth Roofs

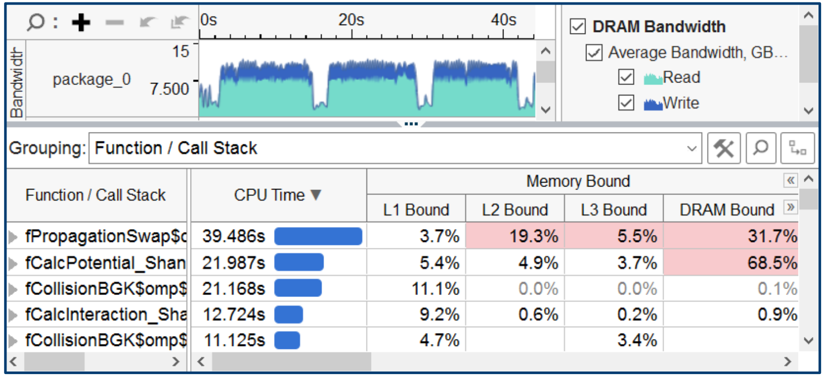


- Memory Access Patterns (MAP) identifies inefficient access patterns.
- Intel® SIMD Data Layout Templates (Intel® SDLT) allows code written as AOS to be stored as efficient SOA.
- Intel® VTune™ Amplifier can be used to further optimize cache usage.
- If cache usage cannot be improved, try re-working the algorithm to increase the AI (and slide up the roof)



Site Location	Strides Distribution	Access Pattern	Max. Site Footprint	Recommendations
[loop in main at roofline.cpp:1...	0% / 100% / 0%	All const strides	38KB	1 Inefficient me...
[loop in main at roofline.cpp:1...	50% / 50% / 0%	Mixed strides	10KB	1 Inefficient me...
[loop in main at roofline.cpp:1...	100% / 0% / 0%	All unit strides	9KB	

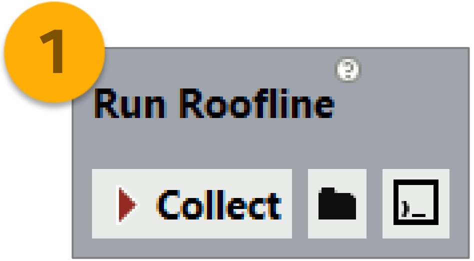
ID		Stride	Type	Source	Variable references	Max. Site Footprint	Access Type
⊞P1		8	Constant stride	roofline.cpp:127	AoS1_Y	38KB	Read
⊞P2		2	Constant stride	roofline.cpp:127	AoS1_X	10KB	Write

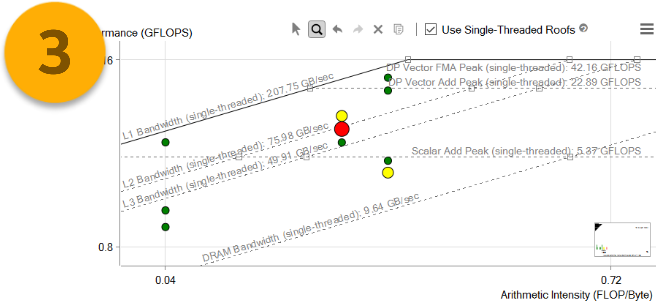


Grouping: Function / Call Stack

Function / Call Stack	CPU Time ▼	Memory Bound			
		L1 Bound	L2 Bound	L3 Bound	DRAM Bound
fPropagationSwap$c	39.486s	3.7%	19.3%	5.5%	31.7%
fCalcPotential_Shan	21.987s	5.4%	4.9%	3.7%	68.5%
fCollisionBGKomp	21.168s	11.1%	0.0%	0.0%	0.1%
fCalcInteraction_Sha	12.724s	9.2%	0.6%	0.2%	0.9%
fCollisionBGKomp	11.125s	4.7%		3.4%	

Summary





- Intel® Advisor's Roofline Chart is highly customizable and easy to generate.
- Identify the best optimization candidates by focusing on low, large loops.
- Use the chart to identify the most likely bottlenecks.
- Intel® Advisor's many other features allow deep analysis of suspected problems and provide advice on how to overcome them.

Legal Disclaimer & Optimization Notice

<Benchmarks/Performance Claims PRIOR TO JAN 3 2018>

[Note the phrase “benchmark results” may be tweaked to say “Estimated results” when estimates are used]

Benchmark results were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown". Implementation of these updates may make these results inapplicable to your device or system.

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. For more complete information visit www.intel.com/benchmarks.

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.

Copyright © 2018, Intel Corporation. All rights reserved. Intel, Pentium, Xeon, Xeon Phi, Core, VTune, Cilk, and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

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

Legal Disclaimer & Optimization Notice

<Benchmarks/Performance Claims AFTER JAN 3 2018>

[Note the phrase “benchmark results” may be tweaked to say “Estimated results” when estimates are used]

The benchmark results reported above may need to be revised as additional testing is conducted. The results depend on the specific platform configurations and workloads utilized in the testing, and may not be applicable to any particular user’s components, computer system or workloads. The results are not necessarily representative of other benchmarks and other benchmark results may show greater or lesser impact from mitigations.

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. For more complete information visit www.intel.com/benchmarks.

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.

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

intel®

Software