ROOFLINE ANALYSIS: A NEW WAY TO VISUALIZE PERFORMANCE OPTIMIZATION TRADEOFFS

Kevin O’Leary, Intel Technical Consulting Engineer
Agenda

- Intel® Advisor Roofline analysis
- Intel® Advisor Demo
- Summary
ROOFLINE MODEL – A VISUALLY INTUITIVE PERFORMANCE MODEL

Combines
- memory utilization/demand
- CPU utilization

Into the same performance analysis and modeling space

\[ AI = \frac{\# \text{ FLOPs}}{\# \text{ BYTES}} \]
What makes loops A, B, C different?
ROOFLINE AUTOMATION IN INTEL ADVISOR 2017 UPDATE 1

Send e-mail to vector_advisor@intel.com to get access to tech preview version of Roofline

Each Roof (slope) gives peak CPU/Memory throughput of your PLATFORM (benchmarked)

Each Dot represents loop or function in YOUR APPLICATION (profiled)

- Interactive mapping to source and performance profile
- Synergy between Vector Advisor and Roofline: FMA example
- Customizable chart
Roofline application profile:

Axis Y: \( \text{FLOP/S} = \#\text{FLOP} \text{ (mask aware)} / \#\text{Seconds} \)

Axis X: \( \text{AI} = \#\text{FLOP} / \#\text{Bytes} \)

Seconds
User-mode sampling
Root access not needed

Performance = Flops/seconds

Roofs
Microbenchmarks
Actual peak for the current configuration

#FLOP
Binary Instrumentation
Does not rely on CPU counters

Bytes
Binary Instrumentation
Counts operands size (not cachelines)
### Getting FLOP/S in Advisor

<table>
<thead>
<tr>
<th>FLOP/S = #FLOP/Seconds</th>
<th>Seconds</th>
<th>#FLOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>- Mask Utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- #Bytes</td>
</tr>
</tbody>
</table>

#### Step 1: Survey
- Non intrusive. *Representative*
- Output: Seconds (+much more)

#### Step 2: Trip counts+FLOPS
- Precise, instrumentation based
- Physically count Num-Instructions
- Output: #FLOP, #Bytes
CACHE-AWARE VS. CLASSIC ROOFLINE

AI = # FLOP / # BYTE

AI_DRAM =

# FLOP/ # BYTES (CPU & Cache ↔ DRAM)

- “DRAM traffic” (or MCDRAM-traffic-based)
- Variable for the same code/platform (varies with dataset size/trip count)
- Can be measured relative to different memory hierarchy levels – cache level, HBM, DRAM

AI_CARM =

# FLOP / # BYTES (CPU ↔ Memory Sub-system)

- “Algorithmic”, “Cumulative (L1+L2+LLC+DRAM)” traffic-based
- Invariant for the given code / platform combination
- Typically AI_CARM < AI_DRAM
**INTERPRETING ROOFLINE DATA**

**Final** Limits
(assuming perfect optimization)
Long-term ROI, optimization strategy

**Current** Limits
(what are my current bottlenecks)
Next step, optimization tactics

**Finally compute-bound**
Invest more into effective CPU/VPU (SIMD) optimization

**Finally memory-bound**
Invest more into effective cache utilization

Not LLC-bound
(DRAM & LLC BW is not #1 bottleneck)
(DRAM BW is not #1 bottleneck)

"Limited by everything"
Check your Advisor Survey and MAP results
What are some of the questions when optimizing/modernizing software?

1. Can we do any better?

2. What are the key performance bottlenecks?
   - Memory subsystem?
   - Lack of CPU/Vectorization/Threading?

3. How much speed-up can I get if I optimize a particular bottleneck?

4. How much speed-up can I get if I optimize (use) another platform?

As the platform gets bigger and more complex – getting these answers is not trivial. ROOFLINE ANALYSIS – is the key to the solution!!
What are my memory and compute peaks?
How far away from peak system performance is my application?

1. Can we do any better?
2. How far are we?

Roofline Ingredient #1
WHERE SHOULD I BE OPTIMIZING?
ROOFLINE ANALYSIS GIVES THE ANSWER

Where is my application?

Where should I be optimizing?
OPTIMIZE IN THE RIGHT PLACE!
ROOFLINE ANALYSIS: SUGGESTED OPTIMIZATION DIRECTIONS

Where you are in the roofline chart suggests your next steps
Advisor roofline “Getting started”

1. Set the env variable ADVIXE_EXPERIMENTAL=roofline
2. Change “Project Properties” to collect FLOPS. Memory traffic and Masks usage.
Advisor roofline “Getting started”

Click on Collect under Roofline

OR
You can also collect both survey and Trip counts
Advisor roofline “Getting started”

Click on Roofline bar to see chart
ROOFLINE AUTOMATION IN INTEL® ADVISOR
Perform the right optimization for your region

Roofline: characterization regions

![Graph showing the Roofline model with different regions and lines representing data throughput limits.]

- **Scalar ~2.3 Peak GFLOP/sec**
- **L1 <-> core 155 Gb/sec**
- **DRAM per core 3 Gb/sec**
- **L1/L2/LLC/DRAM-bound**
  - Investing into Compute peak could be useless
- **L2/LLC/DRAM/Compute-bound**
  - Al == 1
- **Compute-Bound**
  - Investing into Cache/DRAM could be useless
How much optimization is left to do?

1. Can we do any better?

DP FMA ~9 Peak GFLOP/sec

What are the limiting factors?:
- Memory subsystem?
- Lack of CPU/Vectorization/Threading?

3. How much speed-up can I get out of:
   - Optimizing for Memory subsystem?
   - Optimizing for CPU/Vectorization/Threading?
   - Purchasing new hardware?

where to invest, where to start and what is a next step in terms of my code?
NOW AVAILABLE IN INTEL ADVISOR 2017 UPDATE 1

1. Interactive mapping to source and performance profile
2. Synergy between Vector Advisor and Roofline
3. Customizable chart
INTEL® ADVISOR DEMO
Call to Action

- Modernize your Code
  - To get the most out of your hardware, you need to modernize your code with vectorization and threading.
  - Taking a methodical approach such as the one outlined in this presentation, and taking advantage of the powerful tools in Intel® Parallel Studio XE, can make the modernization task dramatically easier.
  - Send e-mail to vector_advisor@intel.com to get the latest information on some exciting new capabilities that are currently under development.
RESOURCES

- **Intel® Advisor Links**
  - Vectorization Guide
  - Explicit Vector Programming in Fortran
  - Optimization Reports
  - Beta Registration & Download

- **Code Modernization Links**
  - Modern Code Developer Community
    - software.intel.com/modern-code
  - Intel Code Modernization Enablement Program
    - software.intel.com/code-modernization-enablement
  - Intel Parallel Computing Centers
    - software.intel.com/ipcc
  - Technical Webinar Series Registration
  - Intel Parallel Universe Magazine
    - software.intel.com/intel-parallel-universe-magazine
For Intel® Xeon Phi™ coprocessors, but also applicable:

- 
- 

Intel® Parallel Studio XE Composer Edition User and Reference Guides:

- 
- 

Compiler User Forums

-
## Configurations for 2007-2016 Benchmarks

### Platform Hardware and Software Configuration

<table>
<thead>
<tr>
<th>Platform</th>
<th>Unscaled Core Frequency</th>
<th>Cores/Sockets</th>
<th>Num Sockets</th>
<th>L1 Data Cache</th>
<th>L2 Cache</th>
<th>L3 Cache</th>
<th>Memory</th>
<th>Memory Frequency</th>
<th>Memory Access</th>
<th>H/W Prefetchers Enabled</th>
<th>HT Enabled</th>
<th>Turbo Enabled</th>
<th>C States</th>
<th>O/S Name</th>
<th>Operating System</th>
<th>Compiler Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® Xeon™ 5472 Processor</td>
<td>3.0 GHZ</td>
<td>4</td>
<td>2</td>
<td>32K</td>
<td>6 MB</td>
<td>None</td>
<td>32 GB</td>
<td>800 MHz</td>
<td>UMA</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Disabled</td>
<td>Fedora 20</td>
<td>3.11.10-301.fc20</td>
<td>icc version 14.0.1</td>
</tr>
<tr>
<td>Intel® Xeon™ X5570 Processor</td>
<td>2.9 GHZ</td>
<td>4</td>
<td>2</td>
<td>32K</td>
<td>256K</td>
<td>8 MB</td>
<td>48 GB</td>
<td>1333 MHz</td>
<td>NUMA</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Disabled</td>
<td>Fedora 20</td>
<td>3.11.10-301.fc20</td>
<td>icc version 14.0.1</td>
</tr>
<tr>
<td>Intel® Xeon™ X5680 Processor</td>
<td>3.33 GHZ</td>
<td>6</td>
<td>2</td>
<td>32K</td>
<td>256K</td>
<td>12 MB</td>
<td>48 MB</td>
<td>1333 MHz</td>
<td>NUMA</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Disabled</td>
<td>Fedora 20</td>
<td>3.11.10-301.fc20</td>
<td>icc version 14.0.1</td>
</tr>
<tr>
<td>Intel® Xeon™ X5570 Processor</td>
<td>2.9 GHZ</td>
<td>8</td>
<td>2</td>
<td>32K</td>
<td>256K</td>
<td>20 MB</td>
<td>64 GB</td>
<td>1600 MHz</td>
<td>NUMA</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Disabled</td>
<td>Fedora 20</td>
<td>3.11.10-301.fc20</td>
<td>icc version 14.0.1</td>
</tr>
<tr>
<td>Intel® Xeon™ X5690 Processor</td>
<td>2.7 GHZ</td>
<td>12</td>
<td>2</td>
<td>32K</td>
<td>256K</td>
<td>30 MB</td>
<td>64 GB</td>
<td>1867 MHz</td>
<td>NUMA</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Disabled</td>
<td>Fedora 20</td>
<td>3.11.10-301.fc20</td>
<td>icc version 14.0.1</td>
</tr>
<tr>
<td>Intel® Xeon™ X5680 Processor</td>
<td>2.2 GHZ</td>
<td>18</td>
<td>2</td>
<td>32K</td>
<td>256K</td>
<td>46 MB</td>
<td>128 GB</td>
<td>2133 MHz</td>
<td>NUMA</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Disabled</td>
<td>Fedora 20</td>
<td>3.13.5-202.fc20</td>
<td>icc version 14.0.1</td>
</tr>
<tr>
<td>Intel® Xeon™ X5690 Processor</td>
<td>2.3 GHZ</td>
<td>18</td>
<td>2</td>
<td>32K</td>
<td>256K</td>
<td>46 MB</td>
<td>256 GB</td>
<td>2400 MHz</td>
<td>NUMA</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Disabled</td>
<td>RHEL 7.0</td>
<td>3.10.0-123.64</td>
<td>icc version 14.0.1</td>
</tr>
<tr>
<td>Intel® Xeon™ X5600v4 Processor</td>
<td>2.2 GHZ</td>
<td>22</td>
<td>2</td>
<td>32K</td>
<td>256K</td>
<td>56 MB</td>
<td>128 GB</td>
<td>2133 MHz</td>
<td>NUMA</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Disabled</td>
<td>CentOS 7.2</td>
<td>3.10.0-327.64</td>
<td>icc version 14.0.1</td>
</tr>
</tbody>
</table>

### 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

Performance measured in Intel Labs by Intel employees.
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.

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.

Copyright © 2016, 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