SNAPSHOT YOUR PERFORMANCE AND IMPROVE!

James Tullos, Technical Consulting Engineer
Intel® Software Development Products
WHY ARE YOU HERE?

- Application performance analysis doesn’t always scale with application size
- Application Performance Snapshot (APS) helps with that challenge
  - Triage performance bottlenecks
  - Determine next steps
- Deep dive with other performance tools based on results
PERFORMANCE OPTIMIZATION WORKFLOW

High Level Overview

- MPI
- CPU
- Memory
- Threading
- Efficiency
- Vectorization

APS
SCALING ANALYSIS CHALLENGES
SCALING ANALYSIS CHALLENGES

Application Growth

System Growth

Problem Growth

Application/Tool Performance

Application Size

- Application
- Analysis Tools
- Scalable Tools
APPLICATION PERFORMANCE SNAPSHOT
APPLICATION PERFORMANCE SNAPSHOT (APS)

- High-level overview of application performance
- Merges previous Application Performance Snapshot with MPI Performance Snapshot
- Identify primary optimization areas and next steps in analysis
- Extremely easy to use
- Informative, actionable data in clean HTML report
  - Also includes recommendations for next steps in analysis
- Detailed reports available via command line
- Scales to large jobs
- Multiple methods to obtain
  - Part of Intel® Parallel Studio XE 2018 Beta
    - Only in Linux* package
  - Separate download from Performance Snapshot page
    - Current version only for Linux*, older version available for Windows*
Application Performance Snapshot

73.19s
Elapsed Time

4.01
CPI
(MAX 4.11, MIN 3.94)

**MPI Time**
15.69% of Elapsed Time (11.48s)

**OpenMP Imbalance**
30.77% of Elapsed Time (22.52s)

**SIMD Instr. per Cycle**
0.09

**Memory Footprint**
AVG 0.00 MB, PEAK 0.00 MB

**I/O Bound**
0.00%
(AVG 0.00, PEAK 0.00)

**FP Instruction Mix**
- % of Packed SIMD Instr.: 27.66%
- % of Scalar SIMD Instr.: 72.34%

Your application has significant OpenMP imbalance. Use OpenMP profiling tools like Intel® VTune™ Amplifier to see the imbalance details.
APS USAGE

Setup Environment
• source /opt/intel/vtune_amplifier_2018/apsvars.sh

Run Application
• mpirun <mpi options> aps.sh <application and args>

Report on Results
• aps-report <options> <result folder>
HTML REPORT BREAKDOWN - OVERVIEW

- Overview shows all areas and relative impact on code performance
- Gives recommendation for next step in performance analysis

Your application has significant OpenMP imbalance. Use OpenMP profiling tools like Intel® VTune™ Amplifier to see the imbalance details.

<table>
<thead>
<tr>
<th>Category</th>
<th>Current</th>
<th>Target</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI Time</td>
<td>15.69%</td>
<td>&lt;15%</td>
<td></td>
</tr>
<tr>
<td>OpenMP imbalance</td>
<td>30.77%</td>
<td>&lt;10%</td>
<td></td>
</tr>
<tr>
<td>Back-End Stalls</td>
<td>82.93%</td>
<td>&lt;20%</td>
<td></td>
</tr>
<tr>
<td>SIMD Instr. per Cycle</td>
<td>0.09%</td>
<td>&gt;1</td>
<td></td>
</tr>
<tr>
<td>I/O Bound</td>
<td>0.00%</td>
<td>&lt;10%</td>
<td></td>
</tr>
</tbody>
</table>
HTML REPORT BREAKDOWN

- **MPI Time** – How much time was spent in MPI calls
- **MPI Imbalance** – Unproductive time spent in MPI library waiting for data

### MPI Time
15.69% of Elapsed Time (11.48s)

### MPI Imbalance
4.59% of Elapsed Time (3.36s)

<table>
<thead>
<tr>
<th>TOP 5 MPI Functions</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waitall</td>
<td>7.47</td>
</tr>
<tr>
<td>Barrier</td>
<td>3.84</td>
</tr>
<tr>
<td>Irecv</td>
<td>2.26</td>
</tr>
<tr>
<td>Isend</td>
<td>1.95</td>
</tr>
<tr>
<td>Scatterv</td>
<td>0.01</td>
</tr>
</tbody>
</table>
HTML REPORT BREAKDOWN

- **OpenMP Imbalance** – Time spent at OpenMP Synchronization Barriers
- **SIMD Instr. per Cycle** – Indicates utilization of FPU
  - **FP Instruction Mix** – shows breakdown of scalar vs. vectorized instructions
**HTML REPORT BREAKDOWN**

- **Back-End Stalls** – indicates percentage of CPU pipeline slots that remain empty
  - **L2 Hit Bound** – Percentage of cycles spent retrieving data from L2
  - **L2 Miss Bound** – Percentage of cycles spent waiting for data beyond L2
APS COMMAND LINE REPORTS - SUMMARY

Summary information

Application: heart_demo.test02
Number of ranks: 8
Used statistics: stat_20170502/
Creation date: 2017-05-02 11:44:27

Your application has significant OpenMP imbalance. Use OpenMP profiling tools like Intel(R) VTune(TM) Amplifier to see the imbalance details.

Elapsed time: 73.19 sec
CPI Rate: 4.01
The CPI value may be too high. This could be caused by such issues as memory stalls, instruction starvation, branch misprediction, or long latency instructions.
Use Intel(R) VTune(TM) Amplifier General Exploration analysis to specify particular reasons of high CPI.

MPI Time: 11.48 sec 15.69%
Your application is MPI bound. This may be caused by high busy wait time inside the library (imbalance), non-optimal communication schema or MPI library settings. Explore the MPI Imbalance metric if it is available or use MPI profiling tools like Intel(R) Trace Analyzer and Collector to explore possible performance bottlenecks.

MPI Imbalance: 3.36 sec 4.59%
OpenMP Imbalance: 22.52 sec 30.77%
The metric value can indicate significant time spent by threads waiting at barriers. Consider using dynamic work scheduling to reduce the imbalance where possible. Use Intel(R) VTune(TM) Amplifier HPC Performance Characterisation analysis to review imbalance data distributed by barriers of different lexical regions.
APS COMMAND LINE REPORTS - COUNTERS

- Shows counters collected during run
- Min, max, and average are across all ranks in run
## APS COMMAND LINE REPORTS – MPI TIME PER RANK

### MPI Time per Rank

<table>
<thead>
<tr>
<th>Rank</th>
<th>LifeTime (sec)</th>
<th>MPI Time (sec)</th>
<th>MPI Time (%)</th>
<th>Imbalance (sec)</th>
<th>Imbalance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0007</td>
<td>72.52</td>
<td>14.31</td>
<td>19.74</td>
<td>4.84</td>
<td>6.67</td>
</tr>
<tr>
<td>0004</td>
<td>72.53</td>
<td>11.57</td>
<td>15.96</td>
<td>3.26</td>
<td>4.50</td>
</tr>
<tr>
<td>0005</td>
<td>72.52</td>
<td>11.40</td>
<td>15.72</td>
<td>3.20</td>
<td>4.42</td>
</tr>
<tr>
<td>0006</td>
<td>72.51</td>
<td>11.11</td>
<td>15.32</td>
<td>3.17</td>
<td>4.37</td>
</tr>
<tr>
<td>0000</td>
<td>72.49</td>
<td>11.08</td>
<td>15.29</td>
<td>4.33</td>
<td>5.97</td>
</tr>
<tr>
<td>0001</td>
<td>72.52</td>
<td>10.95</td>
<td>15.10</td>
<td>3.01</td>
<td>4.15</td>
</tr>
<tr>
<td>0002</td>
<td>72.49</td>
<td>10.79</td>
<td>14.88</td>
<td>2.57</td>
<td>3.55</td>
</tr>
<tr>
<td>0003</td>
<td>72.50</td>
<td>10.64</td>
<td>14.68</td>
<td>2.50</td>
<td>3.45</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>580.07</strong></td>
<td><strong>91.86</strong></td>
<td><strong>15.84</strong></td>
<td><strong>26.88</strong></td>
<td><strong>4.63</strong></td>
</tr>
<tr>
<td><strong>AVG</strong></td>
<td><strong>72.51</strong></td>
<td><strong>11.48</strong></td>
<td><strong>15.84</strong></td>
<td><strong>3.36</strong></td>
<td><strong>4.63</strong></td>
</tr>
</tbody>
</table>
NEXT STEPS IN ANALYSIS
PERFORMANCE OPTIMIZATION WORKFLOW

Application Performance Snapshot

- ITAC
- MPI
- OpenMP

- Am
- CPU
- FPU, SIMD, Backend

- Am
- Memory
- Vectorization

© 2017 Intel Corporation. All rights reserved. Intel and the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.
For more complete information about compiler optimizations, see our Optimization Notice.
FULL MPI PROFILING VIA INTEL® TRACE ANALYZER AND COLLECTOR

- Compare 2 communication profiles – focus on bottlenecks
- Shows how MPI processes interact
Focus on node-level profiling via Intel® VTune Amplifier

- Rank Selection/Multi-selection in ITAC for Profiling in VTune
  - Automatically generate MPI command line script
  - User will run command line script and automatically launch profiling of selected ranks by VTune
**NEW!**

**Improve Performance of MPI Applications**

**Enhanced MPI Metrics for Analysis**

**Threading: CPU Utilization**
- Serial vs. Parallel time
- Top OpenMP regions by potential gain
- Tip: Use hotspot OpenMP region analysis for more detail

**Memory Access Efficiency**
- Stalls by memory hierarchy
- Bandwidth utilization
- Tip: Use Memory Access analysis

**Vectorization: FPU Utilization**
- FLOPS\(^\dagger\) estimates from sampling
- Tip: Use Intel Advisor for precise metrics and vectorization optimization

**MPI Imbalance Metric**
- Metric for performance of rank on critical path
- Computational bottlenecks and outlier rank behavior now available in VTune Amplifier
- For communication pattern problems between ranks use Intel® Trace Analyzer and Collector (ITAC)

\(^\dagger\) For 3rd, 5th, 6th Generation Intel® Core™ processors and second generation Intel® Xeon Phi™ processor code named Knights Landing.
Find Effective Optimization Strategies
Intel Advisor: Cache-aware roofline analysis

Roofs Show Platform Limits
- Memory, cache & compute limits

Dots Are Loops
- Bigger, red dots take more time so optimization has a bigger impact
- Dots farther from a roof have more room for improvement

Higher Dot = Higher GFLOPs/sec
- Optimization moves dots up
- Algorithmic changes move dots horizontally

Which loops should we optimize?
- A and G are the best candidates
- B has room to improve, but will have less impact
- E, C, D, and H are poor candidates
Storage Performance Snapshot
Discover if faster storage can improve server/workstation performance

Learn It On One Coffee Break
- Easy setup
- Quickly see meaningful data
- System view of workload
- Any architecture

Targeted Systems
- Servers & workstations with directly attached storage
- Not scale out storage clusters
- Linux kernel 2.6 or newer dstat 0.7 or newer
- Windows Server 2012, Windows 8 or newer Windows OS

Also included with Intel® Parallel Studio and Intel® VTune™ Amplifier products.
Create Faster HPC, Cloud, and AI Software
What’s New in Intel® Parallel Studio XE 2018 Beta

Get More Performance from New Hardware
- Use fast AVX-512 instructions on Intel® Xeon® and Xeon Phi™ processors
- Accelerate MPI applications with Intel® Omni-Path Architecture support

Discover Untapped Performance Faster
- Intel® Advisor – Use Roofline analysis to find high impact, but under optimized loops
- Application Snapshot – Get quick answers: Does my hybrid code need optimization?
- Intel® VTune™ Amplifier – Profile private clouds with Docker* containers, Java* daemons

Boost Machine Learning Application Performance
- Intel® Data Analytics Acceleration Library – Speed machine learning with new optimized algorithms
- Intel® Distribution for Python® - Accelerate Python code using fast NumPy/SciPy and scikit-learn packages

Latest Standards and IDEs
- C++2017 draft parallelizes and vectorizes C++ easily using Parallel STL*
- Full Fortran® 2008, Fortran 2015 draft
- OpenMP® 5.0 draft, Microsoft Visual Studio® 2017

And much more*…

* See Release Notes for the full list with further updates and new features.

Legal Disclaimer & Optimization Notice

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 © 2017, 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