Fast, light weight, scalable MPI performance analysis

Gergana Slavova

gergana.s.slavova@intel.com

Intel Software and Services Group
Agenda

MPI Scaling Analysis Challenges
Existing tools
MPI Performance Snapshot: Overview & Motivation
Metrics
Beta participation
MPI Scaling Analysis Challenges

- **Scalability** - HPC Cluster Sizes Continue to Grow Faster than Moore's Law Rate
- **Trace File Sizes** - Large Amounts of Data for 100's to 1000's of Ranks
- **Identifying Key Metrics** – Lots of Data and the Need to Identify Key Metrics

**Lightweight & Scalable MPI Analysis Needed**
Lightweight statistics via Intel® MPI Library

- Set I_MPI_STATS to a non-zero integer value to gather MPI communication statistics
- Manipulate the results with I_MPI_STATS_SCOPE to increase effectiveness of the analysis

```bash
$ mpirun -genv I_MPI_STATS 3 -genv I_MPI_STATS_SCOPE coll ...
```

~~~ Process 0 of 16 on node compute-0-0.local
lifetime = 751561.16

Data Transfers

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
<th>Amount(MB)</th>
<th>Transfers</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
<td>0.000000e+00</td>
<td>0</td>
</tr>
<tr>
<td>000</td>
<td>001</td>
<td>1.398373e-02</td>
<td>277</td>
</tr>
<tr>
<td>000</td>
<td>002</td>
<td>1.371384e-02</td>
<td>240</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>6.158352e-02</td>
<td>1031</td>
<td></td>
</tr>
</tbody>
</table>

Communication Activity

<table>
<thead>
<tr>
<th>Operation</th>
<th>Volume(MB)</th>
<th>Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collectives</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allgather</td>
<td>7.629395e-06</td>
<td>2</td>
</tr>
<tr>
<td>Allgatherv</td>
<td>0.000000e+00</td>
<td>0</td>
</tr>
<tr>
<td>Allreduce</td>
<td>1.678467e-03</td>
<td>216</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>2.594000e+00</td>
<td>218</td>
</tr>
</tbody>
</table>

Communication Activity by actual args

<table>
<thead>
<tr>
<th>Operation</th>
<th>Context</th>
<th>Algo</th>
<th>Comm size</th>
<th>Message size</th>
<th>Calls</th>
<th>Cost(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collectives</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allgather</td>
<td>0</td>
<td>4</td>
<td>16</td>
<td>4</td>
<td>1</td>
<td>0.20</td>
</tr>
<tr>
<td>1</td>
<td>96</td>
<td>4</td>
<td>16</td>
<td>4</td>
<td>1</td>
<td>0.21</td>
</tr>
<tr>
<td>Allreduce</td>
<td>1</td>
<td>124</td>
<td>1</td>
<td>16</td>
<td>72</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>124</td>
<td>1</td>
<td>19</td>
<td>8</td>
<td>6</td>
<td>0.13</td>
</tr>
<tr>
<td>3</td>
<td>124</td>
<td>1</td>
<td>16</td>
<td>4</td>
<td>7</td>
<td>0.13</td>
</tr>
<tr>
<td>4</td>
<td>112</td>
<td>1</td>
<td>6</td>
<td>8</td>
<td>1</td>
<td>0.09</td>
</tr>
<tr>
<td>5</td>
<td>112</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>96</td>
<td>1</td>
<td>16</td>
<td>8</td>
<td>200</td>
<td>7.02</td>
</tr>
</tbody>
</table>

| Barrier | 1       | 124  | 2         | 16           | 0     | 2       | 0.01    |
|         | 2       | 108  | 2         | 3            | 0     | 8       | 0.05    |
|

Where is your biggest cost?

Determine where are the communication hotspots

Who’s your biggest consumer?
Get the best of all worlds!

MPI profiling without the overhead

OS & hardware-level counters

Lightweight stats + counter data + “secret” sauce = MPI Performance Snapshot
MPI Performance Snapshot (MPS)

Delivered with Intel® Trace Analyzer & Collector

- Intel has separated statistical analysis from event analysis
- The capability is available now via the command line

New developer capability

- MPS enables the developer to quickly gather and analyze statistics on up to 32,000 ranks
- Shows PAPI counters and MPI/OpenMP load imbalances
- Enables the Intel Trace Analyzer and Collector trace file to be targeted for deeper and focused event-based analysis
Why MPI Performance Snapshot?

Advantages

- Get an initial profile of the application very quickly
- Performance variation at scale can be detected sooner
- Provides development recommendations to developers based on analysis
- Easy to use functionality for out of the box use

Benefits to developers

- Difficult performance issues are easier to spot
- Application performance is improved faster
- Experienced & non-experienced developers can adopt quickly
## Adding High Level Analysis

<table>
<thead>
<tr>
<th></th>
<th>Intel® Trace Analyzer &amp; Collector Core Released Functionality (for Detailed Analysis)</th>
<th>MPI Performance Snapshot (MPS) Functionality (Beta) (for High Level Analysis)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scalability</strong></td>
<td>0-4K ranks</td>
<td>1K – 100K ranks (37K tested)</td>
</tr>
<tr>
<td><strong>Trace Details</strong></td>
<td>High (Events, Source hooks)</td>
<td>Low (aggregation)</td>
</tr>
<tr>
<td><strong>Trace Size</strong></td>
<td>Huge (~17 GB for 1K ranks)</td>
<td>Much less (~0.8 GB for 1K ranks)</td>
</tr>
<tr>
<td><strong>Event-Based Analysis</strong></td>
<td>Yes</td>
<td>Set of PAPI* events</td>
</tr>
<tr>
<td><strong>Statistical Analysis</strong></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Quick Processing</strong></td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Small &amp; Flexible</strong></td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Collector Used</strong></td>
<td>Intel® Trace Collector</td>
<td>Built-in &amp; separate: Intel MPI Statistics + MPI/OMP imbalance</td>
</tr>
</tbody>
</table>

*Other names and brands may be claimed as the property of others.
High Capacity MPI Profiler

Combination of lightweight collector (PAPI events + OMP itt_notify metrics + MPI wait time metrics), internal MPI statistics

Metrics collected now

- MPI time vs application time
- Sum of times spent in every MPI function
- MPI message size and transfer data (all + per rank)
- PAPI* counters: FP instr, Vectorized DP instr, memory access instr
- Memory usage stats (all + per rank)
- MPI and OpenMP imbalance

Currently the only tool (among ones we are aware of) capable of showing OpenMP, MPI imbalance, and PAPI* counters in one place

Easy to use ‘entry point’ for first-time tuners of HPC workloads

Currently Command Line Interface Use on Linux* only
4 quick steps to getting started

1. Install Intel® Trace Analyzer and Collector (ITAC)

2. Setup your environment

   $ source /opt/intel/itac/9.1/bin/mpsvars.sh

3. Run with the MPI Performance Snapshot enabled

   $ mpirun -mps -n 1024 ./exe

4. Analyze your results

   $ mps ./stats.txt ./app_stat.txt
Focus on Memory & Counter Usage

New collector displays summary info immediately after end of application run

HW counters & memory usage info:

```
==================== GENERAL STATISTICS ====================
WallClock:          284.274 sec   (All processes)
MIN:           31.998 sec   (rank 0)
MAX:           35.534 sec   (rank 7)

================== HW COUNTERS STATISTICS ==================
Floating-Point instructions:  45.77%
Vectorized DP instructions:  24.69%
Memory access instructions:  42.35%

================== MEMORY USAGE STATISTICS =================
All processes:    256.740MB
MIN:     30.608MB  (process 7)
MAX:     33.136MB  (process 1)
```
## MPI Imbalance Statistics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
<th>Percentage</th>
<th>Rank/Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI Imbalance</td>
<td>207.847 sec</td>
<td>73.12%</td>
<td>All processes</td>
</tr>
<tr>
<td>MIN</td>
<td>23.044 sec</td>
<td>64.85%</td>
<td>Rank 6</td>
</tr>
<tr>
<td>MAX</td>
<td>30.113 sec</td>
<td>88.57%</td>
<td>Rank 1</td>
</tr>
</tbody>
</table>

## OpenMP Statistics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
<th>Percentage</th>
<th>Regions</th>
<th>Rank/Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenMP Regions</td>
<td>228.631 sec</td>
<td>80.43%</td>
<td>56</td>
<td>All processes</td>
</tr>
<tr>
<td>MIN</td>
<td>25.348 sec</td>
<td>71.33%</td>
<td>7</td>
<td>Rank 7</td>
</tr>
<tr>
<td>MAX</td>
<td>33.124 sec</td>
<td>97.42%</td>
<td>7</td>
<td>Rank 1</td>
</tr>
<tr>
<td>OpenMP Imbalance</td>
<td>103.924 sec</td>
<td>36.56%</td>
<td></td>
<td>All processes</td>
</tr>
<tr>
<td>MIN</td>
<td>11.522 sec</td>
<td>32.43%</td>
<td></td>
<td>Rank 3</td>
</tr>
<tr>
<td>MAX</td>
<td>15.057 sec</td>
<td>44.29%</td>
<td></td>
<td>Rank 2</td>
</tr>
</tbody>
</table>
Easy-to-read HTML output
Helps you categorize performance issues

Application Analysis to Guide Development Efforts
I’ve done the lightweight analysis. What’s next?
Full MPI profiling via Intel® Trace Analyzer and Collector

- **Summary page**
- **Time interval shown**
- **Aggregation of shown data**
- **Tagging & Filtering**
- **Compare**
- **Idealizer**
- **Settings**
- **Imbalance Diagram**

**Compare 2 communication profiles – focus on bottlenecks**

**Shows how MPI processes interact**
Focus on node-level profiling via Intel® VTune™ Amplifier XE

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
Tune Applications for Scalable Multicore Performance
Intel® VTune™ Amplifier XE Performance Profiler

Get the Data You Need
- Hotspot (Statistical call tree), Call counts (Statistical)
- Thread Profiling – Concurrency and Lock & Waits Analysis
- Cache miss, Bandwidth analysis
- GPU Offload and OpenCL* Kernel Tracing on Windows

Find Answers Fast
- View Results on the Source / Assembly
- OpenMP Scalability Analysis, Graphical Frame Analysis
- Filter Out Extraneous Data - Organize Data with Viewpoints
- Visualize Thread & Task Activity on the Timeline

Easy to Use
- No Special Compiles – C, C++, C#, Fortran, Java, ASM
- Visual Studio* Integration or Stand Alone on Windows* or Linux*
- Graphical Interface & Command Line
- Local & Remote Data Collection
- New! Analyze Windows* & Linux* data from OS X*

Quickly Find Tuning Opportunities

See Results On The Source Code

Timeline Visualizes & Filters

1 Events vary by processor.  2 No data collection on OS X*
Intel® Parallel Studio XE 2016 Suites

Vectorization – Boost Performance By Utilizing Vector Instructions / Units

- Intel® Advisor XE - Vectorization Advisor identifies new vectorization opportunities as well as improvements to existing vectorization and highlights them in your code. It makes actionable coding recommendations to boost performance and estimates the speedup.

Scalable MPI Analysis– Fast & Lightweight Analysis for 32K+ Ranks

- Intel® Trace Analyzer and Collector add MPI Performance Snapshot feature for easy to use, scalable MPI statistics collection and analysis of large MPI jobs to identify areas for improvement

Big Data Analytics – Easily Build an IA (Intel® Architecture) Optimized Data Analytics Application

- Intel® Data Analytics Acceleration Library (Intel® DAAL) will help data scientists speed through big data challenges with the use of optimized IA functions

Standards – Scaling Development Efforts Forward

- Supporting the evolution of industry standards of OpenMP*, MPI, Fortran and C++ Intel® Compilers & performance libraries
Resources

Getting Started with the MPI Performance Snapshot


Intel® Trace Analyzer and Collector product page

http://www.intel.com/go/traceanalyzer

Intel® Parallel Studio XE product page (look for Cluster Edition)


Intel® Clusters and HPC Technology forums


Poisson sample code

Available under <top_level_dir>/itac/9.1.0.xxx/examples/poisson
Call to Action

Try it today!

- Provide Critical Feedback on Current MPI Snapshot functionality
- Help Shape Development for the Future Product Release with your Input

Functionality Availability and Requirements

- MPI Performance Snapshot available in Intel® Trace Analyzer and Collector 9.0 Update 3 and the latest Intel® Trace Analyzer and Collector 9.1 Beta
- Intel® MPI Library 5.0 Update 3 required (works with 5.1 Beta)
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 © 2015, 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