Designing Software for Intel Xeon Phi And OmniPath Architecture

Ravindra Babu Ganapathi
  Product Owner/ Technical Lead Omni Path Libraries, Intel Corp.
Sayantan Sur
  Senior Software Engineer, Intel Corp.
All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

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 http://www.intel.com/performance.

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 does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com].

Intel technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

Intel processors of the same SKU may vary in frequency or power as a result of natural variability in the production process.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

© Intel Corporation Intel, the Intel logo, Xeon and Xeon Phi are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.
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 SSE4 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
Agenda

• Intel® Xeon Phi™ x200 Architecture Overview
• Intel® Xeon Phi™ Compiler
• Intel® Xeon Phi™ x200 Software
• Intel® Omni-Path™ Architecture (OPA)
• OPA Transport Layers
• OPA Performance
Intel® Xeon Phi™ x200 Architecture Overview
Knights Landing: Next Intel® Xeon Phi™ Processor

First self-boot Intel® Xeon Phi™ processor that is binary compatible with main line IA. Boots standard OS.

Significant improvement in scalar and vector performance

Integration of Memory on package: innovative memory architecture for high bandwidth and high capacity

Integration of Fabric on package

Enables extreme parallelism with general purpose programming

Potential future options subject to change without notice. All timeframes, features, products and dates are preliminary forecasts and subject to change without further notification.
Knights Landing Overview

Chip: 36 Tiles interconnected by 2D Mesh

Tile: 2 Cores + 2 VPU/core + 1 MB L2

Memory: MCDRAM: 16 GB on-package; High BW DDR4: 6 channels @ 2400 up to 384GB

IO: 36 lanes PCIe Gen3. 4 lanes of DMI for chipset

Node: 1-Socket only

Fabric: Omni-Path on-package (not shown)

Vector: up to 2 TF/s Linpack/DGEMM; 4.6 TF/s SGEMM

Streams Triad: MCDRAM up to 490 GB/s; DDR4 90 GB/s

Scalar: Up to ~3x over current Intel® Xeon Phi™ co-processor 7120 (“Knights Corner”)

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 http://www.intel.com/performance.

Configurations:
1. Intel Xeon Phi processor 7250 (16GB, 1.4 GHz, 68-cores) running LINPACK (score 2000 GFLOPS), DGEMM (score 2070 GFLOPS), SGEMM (4605 GFLOPS), STREAM (DDR4 = 90 GB/s and MCDRAM = 490 GB/s), 96 GB DDR4-2133 memory, BIOS R00.RC085, Cluster Mode = Quad, MCDRAM Flat or Cache, RDMA* 7.0, MPSP 1.2.2, Intel MPI 5.1.2, DGEMM 20K x 20K, LINPACK 100K x 100K size

2. Intel estimates based on <specint-like workloads> comparing configuration 1 to Intel Xeon Phi co-processor 7120A hosted on 2x Intel Xeon processor E5-2697 v3.
**Intel ISA**

**KNL implements all legacy instructions**

**AVX-512 Extensions**
- 512-bit FP/Integer Vectors
- 32 regs, & 8 mask regs
- Gather/Scatter

**Conflict Detection**: Improves Vectorization

**Prefetch**: Gather and Scatter Prefetch

**Exponential and Reciprocal** Instructions

---

1. Previous Code names Intel® Xeon® processors
2. Intel® Xeon Phi™ processor

---

**Segment Specific ISA**

**AVX-512PF**

**AVX-512ER**

---

**Common ISA**

- E5-2600
  - (SNB\(^1\))
  - SSE*
  - AVX
  - x87/MMX
- E5-2600v4
  - (HSW\(^1\))
  - SSE*
  - AVX
  - AVX2
  - BMI
  - AVX-512CD
- 7200
  - (KNL\(^2\))
  - SSE*
  - AVX
  - AVX2
  - BMI
  - AVX-512F
  - AVX-512PF
  - AVX-512ER

---
Knights Landing Products

**KNL**
- DDR4
- PCIe Root port
- 2x16
- 1x4

**KNL with Omni-Path**
- DDR Channels: 6
- MCDRAM: up to 16 GB
- Gen3 PCIe (Root port): 4 lanes
- Omni-Path Fabric: 200 Gb/s/dir

**NTB Chip**
- No DDR Channels
- MCDRAM: up to 16 GB
- Gen3 PCIe (End point): 16 lanes
- NTB Chip to create PCIe EP

**Self Boot Socket**

**PCIe Card**

---

Potential future options subject to change without notice. Codenames. All timeframes, features, products and dates are preliminary forecasts and subject to change without further notification.
KNL Memory Modes

Three Modes. Selected at boot

**Cache Mode**
- 16GB MCDRAM
- DDR
- SW-Transparent, Mem-side cache
- Direct mapped. 64B lines.
- Tags part of line
- Covers whole DDR range

**Flat Mode**
- 16GB MCDRAM
- DDR
- MCDRAM as regular memory
- SW-Managed
- Same address space

**Hybrid Mode**
- 8 or 12GB MCDRAM
- 4 or 8 GB MCDRAM
- Part cache, Part memory
- 25% or 50% cache
- Benefits of both
Flat MCDRAM: SW Architecture

MCDRAM exposed as a separate NUMA node

Memory allocated in DDR by default → Keeps non-critical data out of MCDRAM.

Apps explicitly allocate critical data in MCDRAM. Using two methods:

- “Fast Malloc” functions in High BW library (https://github.com/memkind/memkind)
  - Built on top to existing libnuma API
- “FASTMEM” Compiler Annotation for Intel Fortran

Flat MCDRAM with existing NUMA support in Legacy OS
**KNL Mesh Interconnect – Mesh of Rings**

**Three Cluster Modes:**

1. **All-to-All:** No affinity between Tile, Directory and Memory

2. **Quadrant:** Affinity between Directory and Memory: Default mode. SW transparent

3. **Sub-NUMA Clustering:** Affinity between Tile, Directory, Memory. SW visible

1) L2 miss, 2) Directory access, 3) Memory access, 4) Data return
Many **Trailblazing Improvements** in KNL. But why?

<table>
<thead>
<tr>
<th>Improvements</th>
<th>What/Why</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Self Boot Processor</strong></td>
<td>No PCIe bottleneck. Be same as general purpose CPU</td>
</tr>
<tr>
<td><strong>Binary Compatibility with Xeon</strong></td>
<td>Runs all legacy software. No recompilation.</td>
</tr>
<tr>
<td><strong>New OoO Core</strong></td>
<td>~3x higher ST performance over KNC</td>
</tr>
<tr>
<td><strong>Improved Vector Density</strong></td>
<td>3+ TFLOPS (DP) peak per chip</td>
</tr>
<tr>
<td><strong>New AVX 512 ISA</strong></td>
<td>New 512-bit Vector ISA with Masks</td>
</tr>
<tr>
<td><strong>New memory technology:</strong> MCDRAM + DDR</td>
<td>Large High Bandwidth Memory → MCDRAM</td>
</tr>
<tr>
<td></td>
<td>Huge bulk memory → DDR</td>
</tr>
<tr>
<td><strong>New on-die interconnect:</strong> Mesh</td>
<td>High BW connection between cores and memory</td>
</tr>
<tr>
<td><strong>Integrated Fabric: Omni-Path</strong></td>
<td>Better scalability to large systems. Lower Cost</td>
</tr>
</tbody>
</table>

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 information go to [http://www.intel.com/performance](http://www.intel.com/performance). Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.
Intel® Xeon Phi™ Compiler
Xeon Phi – General Purpose Programming Models

• Significant user-level benefits from using familiar parallel programming models
  • OpenMP 4.x, MPI, TBB, …
  • Single code base for multi-core and many-core CPUs
• Same code optimizations apply to both many-core and multi-core CPUs
• Compiler plays critical role in enabling user applications
Recent Target-specific Compiler Options (ICC)

- `-xMIC-AVX512`: Optimizes code for KNL
- `-xCORE-AVX2`: Optimizes code for HSW
- `-xCORE-AVX512`: Optimizes code for Xeon SKX (Skylake Server)
- `-axMIC-AVX512` or `-axCORE-AVX512`
  - Two versions: baseline and another optimized for KNL or Xeon SKX
  - ‘baseline’: governed by implied `-x` flag, default sse2
  - For each function where the compiler deems there would be a performance benefit by choosing a different code-path, multiple cpu-specific functions will be created by the compiler and execution will go to the correct code-path based on runtime checks
- `-axMIC-AVX512,CORE-AVX2`
  - Three versions: baseline, KNL optimized, HSW optimized
  - `-mmic`: Generate code for KNC
    - Creates binary with different signature than Xeons
Intel® Xeon Phi™ x200 S/w Components

Complete Software release can be found @ https://software.intel.com/en-us/articles/xeon-phi-software
Offload Over Fabric (OOF)

- Offload applications are launched as MPI/PGAS processes on Xeon nodes in clusters
  - Subset: Run offload application on Xeon with set of KNL nodes as target.
- Allocate set of KNL nodes to offload for each MPI/PGAS process on Xeon.
  - 1 MPI/PGAS process on Xeon to N KNL, N>=1
  - OOF uses OFFLOAD_NODES and OFFLOAD_DEVICES (Max 8 target nodes) environment variable to configure offload targets
- Capable of offload from Xeon node to KNL Node(s) over Fabric (Intel OPA or InfiniBand)
  - No code rewrite, Legacy code recompilation required
- Intel® Xeon Phi™ OOF User Guide
Logical grouping of nodes by Resource Manager

Typical Use Case is symmetric offloading. The above is for illustration purpose that asymmetric offload is supported by the software stack.
MPI/PGAS Processes spawned on Xeons
Offload from Xeon to KNL nodes by Offload Application
MCDRAM Library: Memkind & HBWMALLOC API

• Memkind is a heap manager for allocating memory with different properties
  • http://memkind.github.io/memkind/

• A specialized use case of Memkind is HBWMALLOC API
  • High bandwidth memory interface
  • Currently available as stable API
    • http://memkind.github.io/memkind/man_pages/hbwmalloc.html
Intel® Omni-Path™ Architecture
Proven Technology Required for Today’s Bids: Intel® OPA is the Future of High Performance Fabrics

Highly Leverages existing Aries and Intel® True Scale technologies

Leading Edge Integration with Intel® Xeon® processor and Intel® Xeon Phi™ processor

Innovative Features for high fabric performance, resiliency, and QoS

Robust Ecosystem of trusted computing partners and providers

Open Source software and supports standards like the OpenFabrics Alliance*

*Other names and brands may be claimed as property of others.
Intel® Omni-Path Architecture

Accelerating data movement through the fabric

Port latency (includes error detection!)

100-110ns port-to-port latency (with error correction)²

195M messages/sec

160 M messages/sec (MPI message rate uni-directional)

up to 30% higher message rate²

up to 6% higher MPI message rate¹

Intel® Omni-Path 48 port Switch

Intel® Omni-Path HFI adapter

MPI App

MPI App

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 http://www.intel.com/performance.

¹ Based on Intel projections for Wolf River and Prairie River maximum messaging rates, compared to Mellanox CS7500 Director Switch and Mellanox ConnectX-4 adapter and Mellanox SB7700/SB7790 Edge switch product briefs posted on www.mellanox.com as of November 3, 2015.

² Latency reductions based on Mellanox CS7500 Director Switch and Mellanox SB7700/SB7790 Edge switch product briefs posted on www.mellanox.com as of July 1, 2015, compared to Intel measured data that was calculated from difference between back to back osu_latency test and osu_latency test through one switch hop. 10ns variation due to “near” and “far” ports on an Intel® OPA edge switch. All tests performed using Intel® Xeon® E5-2697v3 with Turbo Mode enabled.

* Other names and brands may be claimed as property of others.
CPU-Fabric Integration
with the Intel® Omni-Path Architecture

KEY VALUE VECTORS
✓ Performance
✓ Density
✓ Cost
✓ Power
✓ Reliability

Future Generations
Additional integration, improvements, and features

Future Intel® Xeon® processor (14nm)
Next generation Intel® Xeon Phi™ coprocessor
Intel® Xeon Phi™ processor
Next generation Intel® Xeon® processor
Intel® Xeon® processor E5-2600 v3

TIME

Performance

Density
Cost
Power
Reliability

Tighter Integration
Multi-chip Package Integration
Intel® OPA HFI Card
Intel® OPA
# New Intel® OPA Fabric Features: Fine-grained Control Improves Resiliency and Optimizes Traffic Movement

<table>
<thead>
<tr>
<th>Features</th>
<th>Description</th>
<th>Benefits</th>
</tr>
</thead>
</table>
| Traffic Flow Optimization        | ◦ Optimizes Quality of Service (QoS) in mixed traffic environments, such as storage and MPI  
                                 | ◦ Transmission of lower-priority packets can be paused so higher priority packets can be transmitted | ◦ Ensures high priority traffic is not delayed → Faster time to solution  
                                 |                                                                 | ◦ Deterministic latency → Lowers run-to-run timing inconsistencies |
| Packet Integrity Protection      | ◦ Allows for rapid and transparent recovery of transmission errors on an Intel® OPA link without additional latency  
                                 | ◦ Resends 1056-bit bundle w/ errors only instead of entire packet (based on MTU size) | ◦ Fixes happen at the link level rather than end-to-end level  
                                 |                                                                 | ◦ Much lower latency than Forward Error Correction (FEC) defined in the InfiniBand* specification¹ |
| Dynamic Lane Scaling            | ◦ Maintain link continuity in the event of a failure of one of more physical lanes  
                                 | ◦ Operates with the remaining lanes until the failure can be corrected at a later time | ◦ Enables a workload to continue to completion. **Note:** InfiniBand will shut down the entire link in the event of a physical lane failure |

¹ Lower latency based on the use of InfiniBand with Forward Error Correction (FEC) Mode A or C in the public presentation titled “Option to Bypass Error Marking (supporting comment #205),” authored by Adee Ran (Intel) and Oran Sela (Mellanox), January 2013. Mode A modeled to add as much as 140ns latency above baseline, and Mode C can add up to 90ns latency above baseline. Link: www.ieee802.org/3/bj/public/jan13/ran_3bj_01a_0113.pdf
Intel® Omni-Path Architecture HPC
Design Focus Architected for Your MPI Application

Designed for Performance at Extreme Scale

Applications

I/O Focused Upper Layer Protocols (ULPs)
- Intel® MPI
- Open MPI
- MVAPICH2
- IBM Platform MPI
- SHMEM

Verbs Provider and Driver
- Intel® Omni-Path PSM2

Libfabric

Intel® Omni-Path Host Fabric Interface (HFI)

Intel® Omni-Path Wire Transport

Intel® Omni-Path Enhanced Switching Fabric

Emerging OpenFabrics Interface (OFI) enabled HPC middleware
Intel® OPA: send options

- **Programmed I/O (PIO)**
  - Optimizes Latency and Message Rate for small messages

- **Send DMA (SDMA)**
  - Optimizes Bandwidth for Large messages
  - 16 SDMA Engines for CPU Offload

- **MPI Protocols**
  - Eager – uses PIO and can also use SDMA
  - Rendezvous – uses SDMA
Intel® OPA: receive options

Eager-Receive
- Received data buffers copied to Application Buffer
- No handshake needed
- Used for MPI Eager protocol

Direct Data placement in Application Buffer
- Data placed directly into Application Memory
- Used for Rendezvous protocol

2 Modes of Receiving data Independent of Send mode

Eager

Direct Data Placement

Adapter

Processor

Host Memory

Receive Buffers

CPU

Application RX Buffer

HFI RXE

Header Suppress

Receive Options

In eager receive, received data buffers are copied to the application buffer without the need for a handshake, and it is used for the MPI Eager protocol. Direct data placement in the application buffer involves placing data directly into application memory and is used for the Rendezvous protocol.
Intel® OPA provides efficient data transfer mechanisms

<table>
<thead>
<tr>
<th>Message Size</th>
<th>Send Side</th>
<th>Receive Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to 8KB</td>
<td>PIO Send</td>
<td>Eager Receive</td>
</tr>
<tr>
<td>&gt; 8KB and &lt; 64KB</td>
<td>SDMA</td>
<td>Eager Receive</td>
</tr>
<tr>
<td>64KB or more</td>
<td>SDMA</td>
<td>Expected Receive</td>
</tr>
</tbody>
</table>

- Most efficient data movement method automatically chosen based on message size
- Specific values of thresholds may vary by platform
PSM2 Optimizations - reducing tag queue search time

MPI receive queues (expected and unexpected) are usually searched in a linear manner – this can lead to lengthy search times.

Wild Cards and ordering requirements have complicated search algorithms.

Novel solution in PSM2 to reduce the length of the tag queue that is searched.

Our Idea:

We can distribute queues using a hash of the (TAG, SRC) combination, and ensure that the oldest posted receive matches first to satisfy MPI ordering requirements.
Matching Incoming Messages using Hashing Technique

- Hash both TAG and SRC (no wildcard bucket)
- Hash only TAG (ANY_SRC bucket)
- Hash only SRC (ANY_TAG bucket)
- No Hash (ANY_TAG, ANY_SRC)

Hash function chooses the queue that must be searched in each bucket.
Winner is chosen based on the lowest timestamp value.
While posting receives we know the wildcard, so only one queue can be searched.
Latency, Bandwidth, and Message Rate - Intel® MPI Benchmarks
Intel® Xeon® processor E5-2697A v4
Intel® Omni-Path Architecture (Intel® OPA) - MVAPICH2 2.1

<table>
<thead>
<tr>
<th>Metric</th>
<th>MVAPICH2 2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency (one-way, 1 switch, 8B) [ns] ; PingPong</td>
<td>920</td>
</tr>
<tr>
<td>Bandwidth (1 rank per node, 1 port, uni-dir, 1MB) [GB/s] ; Uniband</td>
<td>12.3</td>
</tr>
<tr>
<td>Bandwidth (1 rank per node, 1 port, bi-dir, 1MB) [GB/s] ; Biband</td>
<td>24.4</td>
</tr>
<tr>
<td>Message Rate (1 rank per node, uni-dir, 8B) [M msg/sec] ; Uniband</td>
<td>4.6</td>
</tr>
<tr>
<td>Message Rate (1 rank per node, bi-dir, 8B) [M msg/sec] ; Biband</td>
<td>5.1</td>
</tr>
<tr>
<td>Message Rate (32 ranks per node, uni-dir, 8B) [M msg/sec]; Uniband</td>
<td>109</td>
</tr>
<tr>
<td>Message Rate (32 ranks per node, bi-dir, 8B) [M msg/sec]; Biband</td>
<td>124</td>
</tr>
</tbody>
</table>

Dual socket servers with one Intel® OPA Edge switch hop. Intel® Xeon® processor E5-2697A v4 2.60 GHz, 16 cores, 2133 MHz DDR4 memory per node. Intel® Turbo Boost Technology enabled, Intel® Hyper-Threading Technology enabled. Intel® MPI Benchmarks 4.1. MVAPICH2 2.1-hfi as packaged with IFS 10.1.1.0.9. Benchmark processes pinned to the cores on the socket that is local to the Intel® OP Host Fabric Interface (HFI) before using the remote socket. RHEL 7.2. BIOS settings: IOU non-posted prefetch disabled. Snoop timer for posted prefetch=9. Early snoop disabled. Cluster on Die disabled.
Latency, Bandwidth, and Message Rate
Intel® Xeon® processor E5-2699 v3 & E5-2699 v4
Intel® Omni-Path Architecture (Intel® OPA)

<table>
<thead>
<tr>
<th>Metric</th>
<th>E5-2699 v3(^1)</th>
<th>E5-2699 v4(^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency (one-way, 1 switch, 8B) [ns]</td>
<td>910</td>
<td>910</td>
</tr>
<tr>
<td>Bandwidth (1 rank per node, 1 port, uni-dir, 1MB) [GB/s]</td>
<td>12.3</td>
<td>12.3</td>
</tr>
<tr>
<td>Bandwidth (1 rank per node, 1 port, bi-dir, 1MB) [GB/s]</td>
<td>24.5</td>
<td>24.5</td>
</tr>
<tr>
<td>Message Rate (max ranks per node, uni-dir, 8B) [Mmps]</td>
<td>112.0</td>
<td>141.1</td>
</tr>
<tr>
<td>Message Rate (max ranks per node, bi-dir, 8B) [Mmps]</td>
<td>137.8</td>
<td>172.5</td>
</tr>
</tbody>
</table>

- Near linear scaling of message rate with added cores on successive Intel® Xeon® processors

Dual socket servers. Intel® Turbo Boost Technology enabled. Intel® Hyper-Threading Technology disabled. OSU OMB 5.1. Intel® OPA: Open MPI 1.10.0-hfi as packaged with IFS 10.0.0.0.697. Benchmark processes pinned to the cores on the socket that is local to the Intel® OP Host Fabric Interface (HFI) before using the remote socket. RHE7.2. Bi-directional message rate measured with osu_mbw_mr, modified for bi-directional measurement. We can provide a description of the code modification if requested. BIOS settings: IOU non-posted prefetch disabled. Snoop timer for posted prefetch=9. Early snoop disabled. Cluster on Die disabled.

1. Intel® Xeon® processor E5-2699 v3 2.30 GHz 18 cores
2. Intel® Xeon® processor E5-2699 v4 2.20 GHz 22 cores
OpenFabrics Interfaces Architecture

Libfabric Enabled Middleware

libfabric
- Control Services
  - Discovery
- Communication Services
  - Connection Management
  - Address Vectors
- Completion Services
  - Event Queues
  - Counters
- Data Transfer Services
  - Message Queues
  - Triggers
  - Tag Matching
  - Atomics

Supported or in active development

Experimental

Intel MPI, MPICH (Netmod), Open MPI (MTL / BTL), Open MPI SHMEM, Sandia SHMEM, GASNet, Clang UPC, rsockets ES-API
OpenFabrics Omni-Path Provider Status

Implements majority of libfabric API by layering on top of PSM2

Performance optimization work is on-going

• Currently adds about 50ns for small messages on top of PSM2 on Xeon core
  • Looking at ways to reduce it further
  • Optimized usage of Active Message interfaces provides good RMA performance for small and large messages
MPICH/CH4 on OFI

The MPICH team is revamping the codebase

- Intel is contributing to the collaboration
- Creating a new lighter-weight channel abstraction (CH4)
- Higher-level semantics offered to netmods that allow better fabric level optimization
- Ability to “inline” critical path code to drastically reduce MPI overheads
- Various memory scalability improvements
- MPICH/CH4 is not yet released, work in progress

### MPICH/CH4/OFI/PSM2 Improvements Compared to MVAPICH2-2.2rc1 on PSM2 on KNL

<table>
<thead>
<tr>
<th>Metric</th>
<th>Percentage Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>osu_latency (8B)</td>
<td>18%</td>
</tr>
<tr>
<td>osu_bw (8B)</td>
<td>22%</td>
</tr>
<tr>
<td>osu_put_latency (8B)</td>
<td>61%</td>
</tr>
<tr>
<td>osu_get_latency (8B)</td>
<td>58%</td>
</tr>
</tbody>
</table>

OSU OMB 5.1 osu_latency, osu_bw, osu_put_latency, osu_get_latency with 1 rank per node. Benchmark processes pinned to the cores on the socket that is local to the Intel® OP Host Fabric Interface (HFI). RHEL 7.2. Both MPICH/CH4/OFI and MVAPICH2-2.2rc1 compiled with gcc (devtoolset-3) and same compiler flags. IFS version 10.2.0.0.134.
Legal Notices and Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Intel, the Intel logo, Xeon and Xeon Phi and others are trademarks of Intel Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others.

© 2015 Intel Corporation.