Intel® MPI Library:
Implementation for Intel® Xeon Phi™
Based Clusters

William Magro
Director & Chief Technologist
Technical Computing Software
Intel Software and Services Group

#OFADevWorkshop
Intel® Xeon Phi™ Coprocessor (codenamed Knights Corner)

Intel® Xeon® Processor

PCIe x16

System Memory

TCP/IP

GDDR5 Channel

GDDR5 Channel

GDDR5 Channel

GDDR5 Channel

GDDR5 Channel

>= 8GB GDDR5 memory

> 50 Cores

Knights Corner

Linux OS
Intel® Xeon Phi™ Coprocessor-based Clusters
Multiple Programming Models

Pthreads, OpenMP*, Intel® Cilk™ Plus, Intel® Threading Building Blocks used for parallelism within MPI processes

MPI Messages
Offload data transfers

New models

Native

Symmetric

* Denotes trademarks of others
Multi- and Many-Core Parallel Programming

**Intel® Xeon® Processor**
- **Intra-node parallel**
  - MKL
  - TBB
  - Cilk Plus
  - OpenMP
  - PThreads

**Intel® Xeon Phi™ Coprocessor**
- **MIC**
  - MKL
  - TBB
  - Cilk Plus
  - OpenMP
  - PThreads

**Intra- and inter-node parallel**
- **Host**
  - Host
- **PCIe**
  - Host to MIC

**Node performance and offload**
- **OpenCL**
  - Host to MIC
- **C++/FTN**
  - Language Extensions for Offload

**SCIF / OFED / IP**

**Maximizes reuse of standard programming models (and code)**
Porting and Running MPI on a Cluster with Intel® Xeon Phi coprocessors

**Standard CPU-based cluster**

<table>
<thead>
<tr>
<th>Build</th>
<th>Run</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ mpiicc -o hello hello.c</td>
<td>$ mpiicc -n 64 -hostfile myhosts ./hello</td>
</tr>
</tbody>
</table>

**Cluster with CPUs and co-processors**

<table>
<thead>
<tr>
<th>Build</th>
<th>Run(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ mpiicc -o hello hello.c</td>
<td></td>
</tr>
<tr>
<td>$ mpiicc -mmic -o hello.mic hello.c</td>
<td>$ export I_MPI_MIC=on</td>
</tr>
<tr>
<td>$ export I_MPI_MIC_POSTFIX=.mic</td>
<td>$ mpirun -n 64 -hostfile myhosts ./hello</td>
</tr>
</tbody>
</table>

```
myhosts:
host1
host2
```

```
myhosts:
host1
host2-mic1
```
Inter-node communication

- **Available Fabrics**
  - Sockets (TCP/IP)
  - Fast Fabrics: DAPL, OFA, TMI

- Fast fabrics access through OFA APIs
- PCIe connection is accessed through OFA APIs (IBSCIF virtual network)

**Multiple DAPL providers:**

**Intel® Xeon Phi™ Coprocessor Communication Link Direct (CCL-direct)**
- Direct access to InfiniBand HW
- Lowest latency data path
- All network segments available

**RDMA over SCIF (IBSCIF)** – RDMA over PCIe (host and its coprocessors)
- High bandwidth data path inside one server

**CCL-proxy** is hybrid provider, pipes CCL-direct and IBSCIF
- Higher bandwidth data path
- All network segments available but especially effective on cross-box communications
Multiple Communication Path Support

**Approach:**
- Select the best fabric for every given network segment, depending on message size
- Large messages transfer utilizes two providers simultaneously
Summary

- Combination of several technologies allows direct programming of Intel® Xeon Phi™ coprocessor based systems as heterogeneous clusters:
  - LSB-compliant Linux* OS on coprocessors
  - Abstraction of PCIe* connection as OFA fabric
  - Direct access to fast fabrics from coprocessor
  - Simultaneous support for multiple fabrics in Intel® MPI Library

* Denotes trademarks of others
Thank You
Intel® MPI Library Overview

- Provides optimized MPI application performance and flexible tuning process
- Delivers industry leading performance and multi-vendor interoperability
- Allows scalability beyond 120K processes
- Supports seamless interoperability with Intel® Trace Analyzer and Collector.

Intel® MPI Library for Intel® Xeon Phi™ coprocessor clusters:

- Direct port to LSB-based SW stack
- Coprocessor as autonomous node
- Heterogeneous cluster with CPU- and coprocessor-based network nodes
- First support in version v4.1
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 © 2012, Intel Corporation. All rights reserved. Intel, the Intel logo, Xeon, Xeon Phi, Xeon Phi logo, Core, VTune, and Cilk are trademarks of Intel Corporation in the U.S. and other countries. *Other names and brands may be claimed as the property of others.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families: Go to:

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