Intel® Xeon Phi™ Coprocessor in Clusters

Respect programming models
Manage Intel® Xeon Phi™ coprocessors in your Clusters for enhanced user experience
What you are going to do...

A bit of theory

Demo – what is possible

3 Steps for the Administrator

Startup scripting

Integration with batch systems
The Intel® Xeon Phi™ Coprocessor
Definitions

Host
- OS
- kernel

Coprocessor
- uOS
- uOS kernel

Intel MPSS stack
- Host side driver
- uOS
- uOS Kernel + drivers
Architecture of the Coprocessor uOS
The Goal – A Multi-Node Cluster
Offload MPI Model

- MPI communication is taking place only between the host processors.
- The coprocessors are used exclusively through offload capabilities.

- Use with Intel® C, C++, and Fortran Compiler for Intel® MIC Architecture, Intel® Math Kernel Library (MKL), etc.
- Using MPI calls inside offloaded code is not supported
Symmetric MPI Model

- Both the host CPUs and the coprocessors execute MPI processes and take part in MPI communication.
- Message passing is supported between all partners using shared memory, TCP or InfiniBand.

- The best fabric is chosen automatically.
- Use with Intel® C, C++, and Fortran Compiler for Intel® MIC Architecture, Intel® Math Kernel Library (MKL), etc.
Next Intel® Xeon Phi™ Processor: Knights Landing

- Designed using Intel’s cutting-edge 14nm process
- Not bound by "offloading" bottlenecks
- **Standalone CPU** or PCIe coprocessor
- Leadership compute & memory bandwidth
- **Integrated** on-package memory
Coprocessor-Only Model Using MPI

• MPI communication is taking place only between the coprocessors.

• The host processors are not used.

• Supported with Intel® C, C++, and Fortran Compiler for Intel® MIC Architecture, Intel® Math Kernel Library (MKL), etc.

• Host is still needed for network traffic
Intel’s Endeavour configuration

- 360 core nodes
- current technology
- 64 GB RAM
- 600 GB SAS HD

- 100+ nodes
- next-gen + prev gen hardware
- 64 GB RAM
- 800 GB SAS HD

Networks:
- 1GEth
- FDR IB

Storage:
- Panasas
- /home
- long-term storage
- DDN Lustre
- LFS9 (SSD)

Nodes:
- admin1
- admin2
- pbs-serv1
- pbs-serv2
- login
- compile
Software Available On Endeavour

Installed in /opt/crtdc/micgnu/VERSION

Path identical on µOS and HOST Linux

To use, set native PATH and LD_LIBRARY_PATH:

```bash
export PATH=/opt/crtdc/micgnu/VERSION/bin:$PATH
export LD_LIBRARY_PATH=/opt/crtdc/micgnu/VERSION/lib64/
```

Python-2.6.6  gdb  openssl-1.0.0
autoconf-2.63  gettext-0.17  perl-5.10.1
bison-2.4.1  glib-2.22.5  pkg-config-0.23
byacc-20070509  libffi-3.0.5  readline-6.0
elfutils-0.152  libhugetlbfs-2.8  rsyslog-5.8.10
expat-2.0.1  ltrace-0.5  strace-4.5.19
file-5.04  lustre-1.8.5  tcl8.5.7
findutils-4.4.2  m4-1.4.13  tcsh
flex-2.5.35  ncurses-5.7  zlib-1.2.3
gawk-3.1.7  ntp-4.2.4p8
gdbm-1.8.0  openssh-5.3p1
File Systems For Intel® Xeon Phi™ Coprocessor

NFS: native support

Lustre*: support provided by Intel

Panasas*: contact vendor to get driver

Mounted both on HOST and Intel® Xeon Phi™ Coprocessor on Endeavour:

/home          NFS
/lfs/lfs6       Lustre
/lfs/lfs7       Lustre
/panfs/home     Panasas
/panfs/panfs2/home1  Panasas
/panfs/panfs2/home2  Panasas
Demo – Symmetric MPI Model
DEMO - Preparing for MPI Tests

Set environment

. /opt/intel/impi/latest/bin64/mpivars.sh
. /opt/intel/compiler/latest/bin/iccvars.sh intel64

Enable coprocessor support:
export I_MPI_MIC=enable
export I_MPI_MIC_POSTFIX=.mic

Prepare hostfile

echo `hostname`-mic0 > hostfile

echo vesg145-mic0 >> hostfile

echo `hostname` >> hostfile

echo vesg145 >> hostfile
DEMO - Run First Test

Compile test program

```bash
cp -r /opt/intel/impi/latest/test/test.c .
mpiicc -o mpitest test.c
mpiicc -mmic -o mpitest.mic test.c
```

Run hello_world test

```bash
mpirun -bootstrap ssh -perhost 1 -f hostfile -n 4 -env I_MPI_DEVICE ssm ~/mpitest
```
DEMO - Run OSU Test

Compile OSU test:

```
cp /home/common/osu* .
mpiicc -o osu_latency osu_latency.c
mpiicc -o osu_latency.mic -mmic osu_latency.c
```

Run OSU test between cards:

```
mpirun -bootstrap ssh -perhost 1 -f hostfile -n 2
          -env I_MPI_DEVICE ssm ~/osu_latency
```

Run OSU test natively over InfiniBand

```
mpirun -bootstrap ssh -perhost 1 -f hostfile -n 2
          -env I_MPI_DEVICE rdma:ofa-v2-mlx4_0-1u
          ~/osu_latency
```
User experience

Many MPI programs can be compiled and run out of the box
Optimization for Intel® Xeon Phi™ architecture still necessary!

But:

- common tools (Compiler, VTune™ Amplifier, libraries) on both platforms
- Similar architecture
- optimization for Intel Xeon Phi architecture will almost always also improve Intel® Xeon® processor code
Step I for administrator
Installation
Recompile HOST Side Drivers

Every cluster is different!

Standard Linux* distribution CD images are almost always out of date (most often security issues)
  -> usually updated kernels are used

Rebuild is easy if all necessary software is installed

Missing software is announced before rebuild is started

rpmbuild --rebuild mpss-modules-* .src.rpm
Quick steps to a running coprocessor

Load the driver: `modprobe mic`

Start the mpss demon: `service mpss start`

- Checks uOS kernel image
- Creates new uOS initrd
- Hands over files to mic.ko driver

Initialize configuration: `micctrl --initdefaults`

Adapt configuration found in `/etc/mpss/` and `/var/mpss/`

Boot the card: `micctrl -rw mic0; micctrl -bw mic0`

Connect via minicom (`/dev/ttyMIC0`)
InfiniBand support

Mellanox ConnectX and Intel Truescale are supported

Various options for software: OFED 1.5.4.1, OFA 3.5-2-mic, OFA 3.12-1, Mellanox* Ofed 2.3; Intel IFS 7.2 ....

In some cases you need to replace driver package (delivered with MPSS) and recompile for kernel used (please see mpss documentation)
Step II for administrator
Ethernet and InfiniBand
Flat Cluster Model

All HOSTs and all Intel® Xeon Phi™ coprocessors can directly communicate with each other. This is enabled by using Linux BRIDGED ETHERNET interfaces.
HOST Setup For Bridged IP

**Redhat* 6 EL ifcfg-br0 file:**

```bash
# cat /etc/sysconfig/network-scripts/ifcfg-br0
DEVICE=br0
TYPE=Bridge
ONBOOT=yes
DELAY=0
NM_CONTROLLED="no"
MTU=9000
BOOTPROTO= dhcp
NOZEROCONF=yes
IPV6INIT=no
```

**Redhat* 6 EL ifcfg-eth0 in bridged mode:**

```bash
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
BRIDGE=br0
MTU=9000
```
Bridged Networking for mpss – host

Bind mic0: on the host to br0:

Change /etc/sysconfig/network-scripts/ifcfg-mic0
DEVICE="mic0"
ONBOOT=yes
BRIDGE=br0
MTU=9000
NM_CONTROLLED="no"

Restart networking

Check network mic0: ifconfig mic0

```
    UP BROADCAST RUNNING MTU:9000 Metric:1
    RX packets:6 errors:0 dropped:0 overruns:0 frame:0
    TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:468 (468.0 b) TX bytes:6110 (5.9 KiB)
```
Set Up Bridged Networking - uOS

Find correct IP ADDRESS for mic0

Modify the uOS Ethernet configuration
/var/mpss/mic0/etc/network/interfaces

...

# The loopback interface
auto lo
iface lo inet loopback
# MIC virtual interface
auto mic0
iface mic0 inet static
  address ADDRESS
  netmask MASK
  gateway GW_ADDRESS
Get InfiniBand running

Procedure:

- service openibd start
- micctrl -rw mic0
- micctrl -bw mic0
- micctrl -s mic0
- service ofed-mic start
- service mpxyd start
Step III for administrator
Cluster Integration
Mount NFS File System On The uOS

No difference from standard Linux*

Be careful with locking and NFS versions (read MPSS documentation)

Keep /etc/hosts on the coprocessor up to date!

Example:
```bash
ssh `hostname`-mic0
mount -o udp,nolock,nfsvers=3 server:/home /home
```
Add Your User To The uOS

Multiple directory systems supported by MPSS

but Linux command `id` is sometimes preferable

Some helpful commands

id -g $AU
id -u $AU

Now used in 3 commands (beware quotes):

```
ssh `hostname`-mic0 "echo '$AU:NOTVALID:`id -u $AU`:`id -g $AU`::/home/$AU:/bin/bash' >> /etc/passwd"
ssh `hostname`-mic0 "echo '$AU:x:`id -g $AU`:' >> /etc/group"
ssh `hostname`-mic0 "echo '$AU:NOTVALID:15392::::::' >> /etc/shadow"
```
Put it all together into a startup script
Startup Script – minimal example

#!/bin/sh

# parameter is a user name
AU=$1

micctrl -rw mic0
micctrl -bw mic0

scp /etc/hosts `hostname`-mic0:/etc/hosts
ssh `hostname`-mic0 "echo '$AU:NOTVALID:`id -u $AU`:`id 
   -g $AU:`/home/$AU:/bin/bash' >> /etc/passwd"
ssh `hostname`-mic0 "echo '$AUSER:x:`id -g $AUSER`:' >> /etc/group"
ssh `hostname`-mic0 "echo '$AUSER:NOTVALID:15392:::' >> /etc/shadow"

mount -o udp,nolock,nfsvers=3 server:/home /home
mkdir /opt
mount -o udp,nolock,nfsvers=3 server:/opt /opt
ssh `hostname`-mic0 killall coi_daemon
ssh `hostname`-mic0 "nohup /usr/bin/coi_daemon --coiuser=$AUSER \
   2>/dev/null >/dev/null &"
Integration Into Batch System

• Most clusters run a batch system
• Most clusters run a prologue/job/epilogue structure
• Prologue/Epilogue are a perfect place to prepare the Intel® Xeon Phi™ coprocessor for each user
• Options at submit time can change environment in a controlled manner
  => COMPLETE separation of user/root
Example from Endeavour

• users access Intel® Xeon Phi™ coprocessors with standard privileges using direct and passwordless ssh

• the home NFS server is mounted, as well as Lustre* and Panasas* shares

• use of bridged networking avoids problems with routing

• number of Intel Xeon Phi coprocessors is automatically detected via lspci
Example from Endeavour continued

• USER is added to all Intel® Xeon Phi™ coprocessor cards on the system, but no password is set
• inetd running by default on the Intel Xeon Phi coprocessor card is removed from init for security reasons
• MTU and NETMASK on the Intel Xeon Phi coprocessor card interface mic0 are corrected
• coi_daemon started as USER
• ssh environment enhanced with ulimits
• OFED MIC-Direct is started automatically
Resources

Intel® Xeon Phi™ Coprocessor developer site: http://software.intel.com/xeonphi
- Guides, Tools and Downloads
- Training (videos, webinars, code samples, books, ...)
- Case Studies, code recipes

Community forum to answer questions

Intel® Xeon Phi™ Coprocessor Cluster Configuration and accompanying zip file with code and configuration files

Resources II

Intel® Xeon Phi™ user guide:  

*Intel® Xeon Phi™ Coprocessor High Performance Programming*  
by Jim Jeffers, James Reinders – introduction to Intel® Xeon Phi™ programming

*High Performance Parallelism Pearls* by Jim Jeffers, James Reinders – 27 showcase examples
Q&A
Backup
Notes On SSH Key Pairs

Typical ssh files:  
```bash
ls ~/.ssh
authorized_keys authorized_keys2 config
id_dsa id_dsa.pub id_rsa id_rsa.pub
id_rsa_ecr identity identity.pub known_hosts
```  

SSH keys are generated by:  
```bash
ssh-keygen -t {rsa|dsa}
```  

For password-less login at least one `~/.ssh/*pub` file needs to be present in `/var/mpss/mic0/root/.ssh/authorized_keys2`:  
```bash
grep "`cat ~/.ssh/id_dsa.pub`"
/var/mpss/mic0/root/.ssh/authorized_keys2
```

In mpss 2.1 you need to use `id_rsa.pub` and  
`/opt/intel/mic/filesystems/mic0/root/.ssh/authorized_keys`
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 © 2014, 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