HIGH PERFORMANCE MULTIPROCESSING COMPUTER
HPMC # 1.00

DISCLAIMER

TCND, Inc. makes no representation or warranties with respect to the contents hereof, and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, TCND, Inc. reserves the right to revise this publication, or our software, and to make changes from time to time in the context hereof, without obligation of TCND, Inc. to notify any person of such revisions or changes.

Contents

HPMC USER GUIDE

17 June 2022

HPMC User Guide consists of five chapters and eight appendices:

Bassem W. Jamaleddine
New York, NY
June 2022

Contents

HPMC USER GUIDE
HPMC 1.00

PREFACE

17 June 2022

Congratulations on your selection of the HPMC calculator:
HPMC = High Peformance Multicore Computer, it is powered by the Linux operating system. This HPMC-calculator is the first in digital computation technology that brings high performance multicores calculation to your desktop.

In this user guide we refer to the UNIX shell prompt with a hash as shown here

```
#  systemctl status mpss
```

Although this manual is made public in June 2022, it forms the pages of many system integrations that I worked on between 2015 and 2022 to build a desktop that can serve as a High Performance Multicore Calculator (HPMC). A desktop that is similar to the Personal Computer (PC) of the 80s: a computer that you can put on the desk. Eventually this desktop ended up as a super-computer that is cased in a chassis whose dimensions are 14 x 14 x 8 inches.

- 2015 six months building the "CUDA CUBE" for CUDA GPU programming
- 2017 six months building the "CUDA Desktop" for CUDA GPU deep learning programming
- 2019 six months building the HPMC adding Intel the Xeon Phi MIC (Many Integrated Cores)
- 2022 six months building the HPMC-KLIB adding the Infiniband for fabric interconnectivity over RDMA

Bassem W. Jamaleddine
New York, NY
June 2022

Contents

Chapter 1
HPMC Calculator

Image File

The HPMC Calculator is a small desktop appliance that is capable of running many threads and many cores for high performance computing. The computer is easy to program without the help of any system administrator. The HPMC is the smallest super computer that can be put on the top of your desk. Its computation power can be compared to the large computers that are farmed in large data center. Unlike the super computers that require tense system administration, the HPMC is treated like a calculator. You will power it on and start exploring with high performance computing - similar to what we used to do on the 80's desktop. Inside your HPMC are two cards: an NVIDIA GPU, and an Intel Xeon Phi. A third card, the InfiniBand HCA adapter may be available on a more advanced HPMC calculator. Hence various technology are embedded inside the HPMC, and they are all being driven to provide you with: CUDA Pogramming, Many Integrated Cores, and a DMA connectivity across multiple HPMCs. The aim of this manual is to introduce you to super computing on your HPMC. It also serves as a technical documentation to show the configuration layers of the devices in your HPMC.

Image File

Read Chapter 1
Back to Contents

1-1 HPMC Calculator

The HPMC may include one or more of the following:

- Xeon Phi Co-Processor called a MIC coprocessor. MIC stands for Many Integrated Cores.
- GPU that is configured for CUDA programming.
- Infiniband fabric connectivity for RDMA. RDMA stands for Remote Direct Memory Access.

Figure 1.1.1 shows the HPMC that includes all the three devices mentioned above. It is the smallest most powerful desktop computer; **dimensions 14 x 14 x 8 inches**.

Figure 1.1.2 shows the side view of the HPMC-KLIB.

Figure 1.1.3 shows the Xeon Phi MIC.

Figure 1.1.4 shows the liquid cooling for the the Xeon Phi MIC.

Figure 1.1.5 shows the QLogic QLE7340 Infiniband QDR slotted in PCIe.

Figure 1.1.6 shows the Fiber optic connected to QLogic InfiniPath QLE7140 IB6110401-01 InfiniBand SDR 4x 10Gb PCI-e x8.

Figure 1.1.7 shows the Network Switch Qlogic 12200-18 18-Port QDR InfiniBand Network Switch 12200-18-28.

Figure 1.1.8 shows the connectors to the Network Switch.

Figure 1.1.9 shows the Xeon Phi Knight Landing 7250 on the main board.

Figure 1.1.10 shows an early HPMC built in 2011.

full view

Figure. 1.1.1 [HPMC-KLIB the smallest most powerful High Performace Multicore Computer]

HPMC 2022 by Bassem Jamaleddine

full view

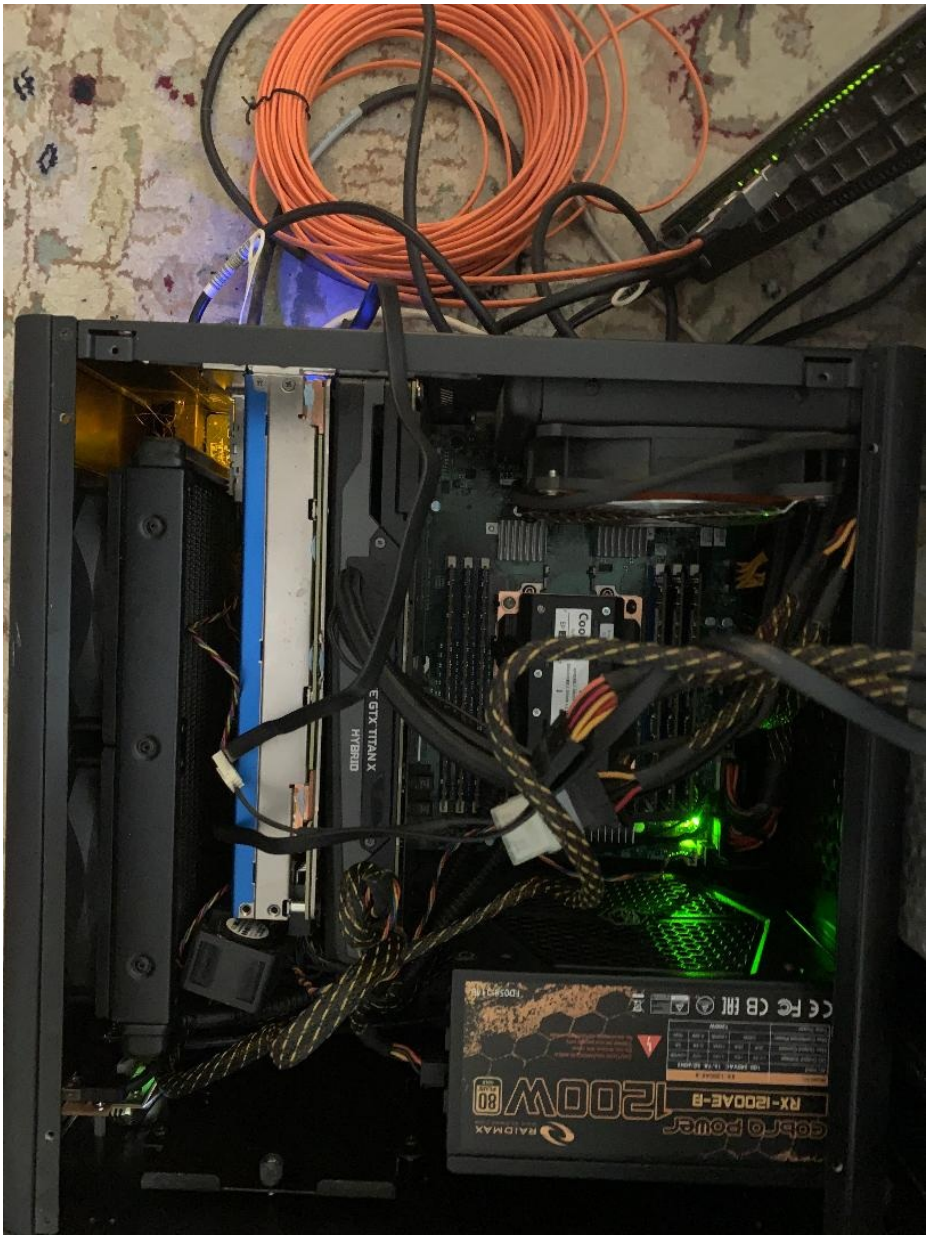


Figure. 1.1.2 [HPMC-KLIB Max's High Performace Multicore Computer]

HPMC 2022 by Max Jamaleddine

full view

Figure. 1.1.3 [Intel Xeon Phi Coprocessors]

HPMC 2022 by Bassem Jamaleddine

full view

Figure. 1.1.4 [Liquid Cooling for the Xeon Phi Coprocessors]

HPMC 2022 by Bassem Jamaleddine

full view

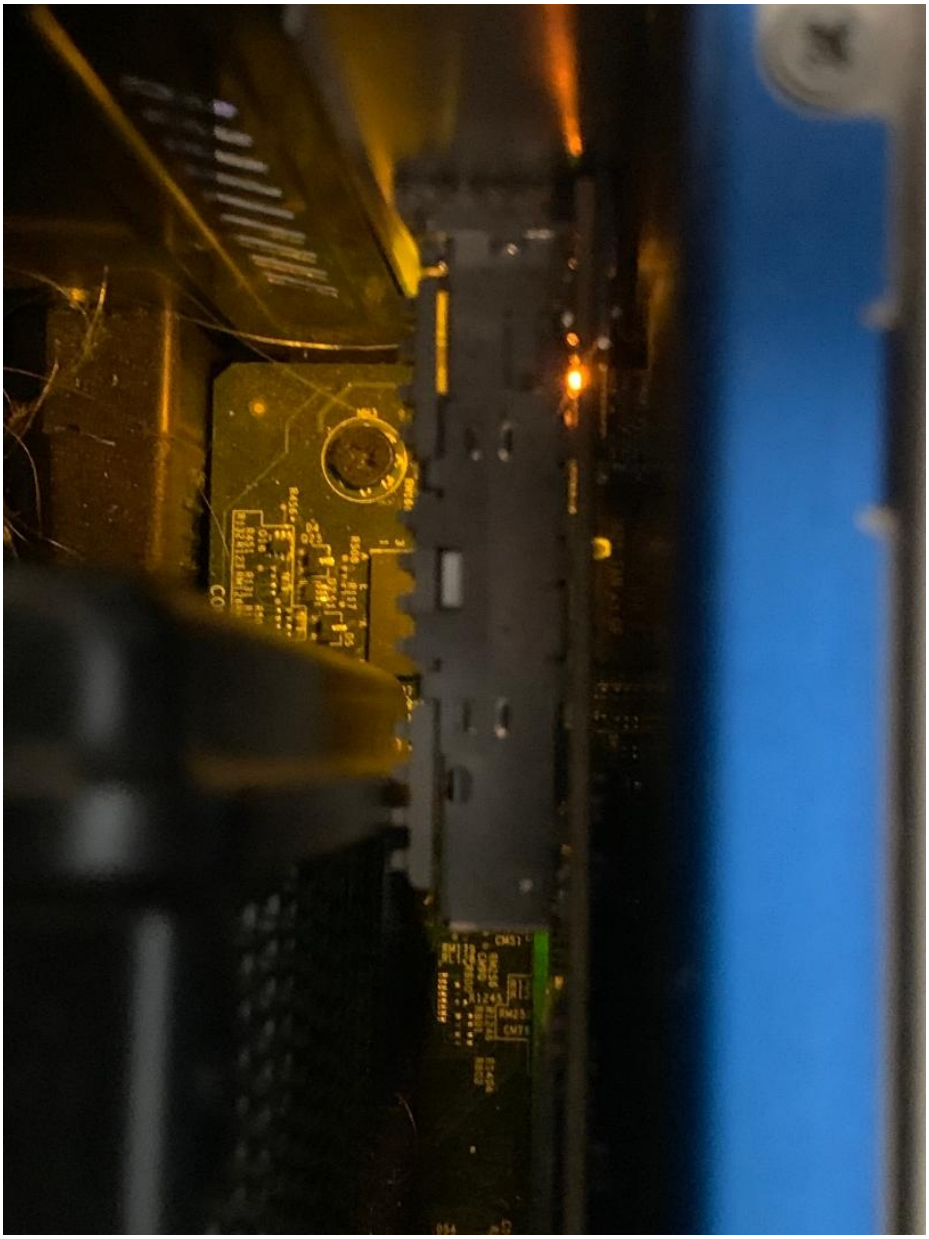


Figure. 1.1.5 [QLogic QLE7340 Infiniband QDR PCIe]

HPMC 2022 by Bassem Jamaleddine

full view

Figure. 1.1.6 [Fiber Optic Connectors in Card]

HPMC 2022 by Bassem Jamaleddine

full view

Figure. 1.1.7 [High Performance Switch (HPS) Network Switch Qlogic]

HPMC 2022 by Bassem Jamaleddine

full view

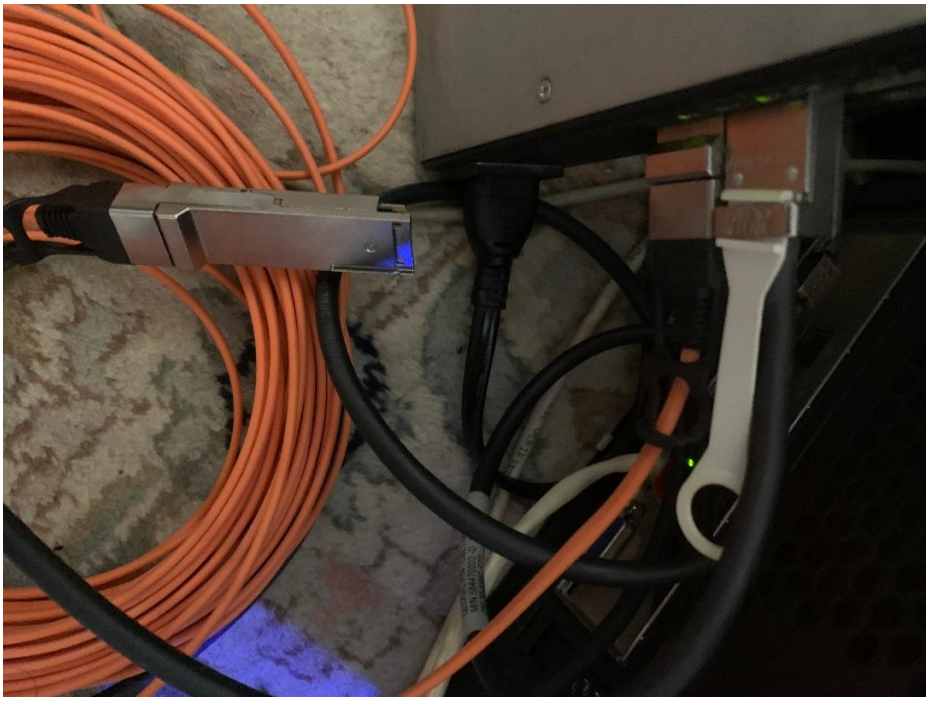


Figure. 1.1.8 [Fiber Connectors in Switch]

HPMC 2022 by Bassem Jamaleddine

full view

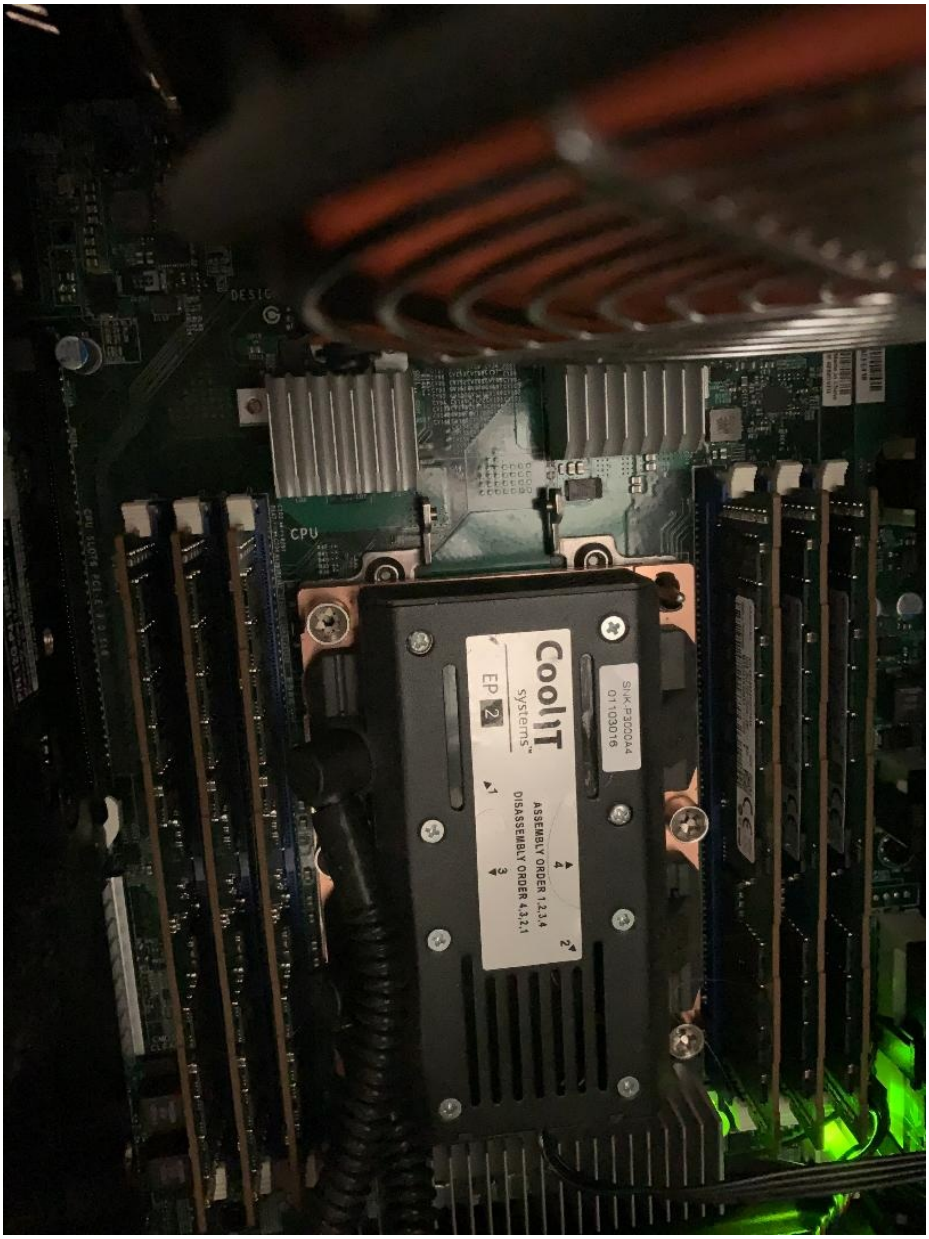


Figure. 1.1.9 [Xeon Phi 7250 on the Main Board]

HPMC 2022 by Bassem Jamaleddine

Figure 1.1.10 shows the MATX, an earlier HPMC with interconnected nodes that I built in 2011.

It is administered through a terminal and runs a custom based REXX like interpreter that has binding to Intel 11 ifort.

full view

Figure. 1.1.10 [MATX 2011 with Inteconnected Nodes]

HPMC 2022 by Bassem Jamaleddine

Chapter 2
Xeon Phi MIC

Image File

Xeon Phi MIC allows you to leverage x86 architecture (CPU with many cores) but allow for more compute throughput by dedicating much of the silicon to floating point ops. The MIC is cache coherent and it increases floating-point throughput. It also uses wide SIMD registers for more throughput. The MIC has fast GDDR5 memory integrated on the card. Later model, like the Knights Landing, features high-bandwidth on-package memory (MCDRAM) to reduce bottlenecks.

The MIC uses numerous low-power cores (60+) to deliver high parallel performance. Unlike GPUs, it supports standard x86 instruction sets, allowing the use of familiar languages like C++ and Fortran along with programming models such as OpenMP, MPI, and Pthreads.

The MIC can operate in three modes: Coprocessor/Offload (offloading tasks from the host CPU), Native (running as a standalone node with its own Linux OS), and Symmetric (using host and card together).

Image File

Read Chapter 2
Back to Contents

2-1 Configuring Xeon Phi MIC

The HPMC is equipped with a Xeon Phi MIC card, called a MIC coprocessor, that is plugged in its bus via the PCIe slot. This coprocessor is memory mapped device, and the motherboard is capable to manage such a memory mapped device on MMIO above 4GB.

Unlike legacy math coprocessor, the MIC is a device that runs its own Linux operating system and can be viewed as an embedded system that is plugged on the bus of the HPMC computer. The MIC uses the ring topology. Refer to LAN/WAN Network Programming with C for more information on the ring topology.

To manage the MIC the following package contains the required modules: **mpss-modules-3.10.0-693.17.1.el7.x86_64-3.8.4-1.x86_64**. This package includes the following:

```
/etc/modprobe.d/mic.conf
/etc/sysconfig/modules/mic.modules
/etc/udev/rules.d/50-udev-mic.rules
/lib/modules/3.10.0-693.17.1.el7.x86_64/extra/mic.ko

/etc/modprobe.d/mic.conf
/etc/sysconfig/modules/mic.modules
/etc/udev/rules.d/50-udev-mic.rules
/lib/modules/3.10.0-693.17.1.el7.x86_64/extra/mic.ko
```

Since the MIC card is a coprocessor that is made available on a host computer as an **embedded Linux system**, it needs to be booted as a computer. For this to happen:
- the MIC device driver must be loaded, hence to communicate and control the MIC hardware layer: lsmod | grep mic
- the MIC is recognized as a device to the host computer: ls /dev/mic0
- the MPSS software layer must be available on the host computer. If it has been installed then: which micinfo

To find information about your HPMC computer motherboard:

```
# dmidecode -t 2
```

```
# dmidecode -t memory
```

```
# cat /proc/iomem
```

```
93000000-b5ffffff : PCI Bus 0000:16
  b5e00000-b5efffff : PCI Bus 0000:17
    b5e00000-b5e1ffff : 0000:17:00.0
      b5e00000-b5e1ffff : mic
```

```
# lspci | grep -i Phi
```

```
17:00.0 Co-processor: Intel Corporation Xeon Phi coprocessor SE10/7120 series (rev 20)
```

```
# lspci -s 17:00.0 -t
```

```
-+-[0000:17]---00.0
 \-[0000:00]-
```

```
# cat /proc/iomem
```

```
93000000-b5ffffff : PCI Bus 0000:16
  b5e00000-b5efffff : PCI Bus 0000:17
    b5e00000-b5e1ffff : 0000:17:00.0
      b5e00000-b5e1ffff : mic
```

lspci -s 17:00.0 -v

```
17:00.0 Co-processor: Intel Corporation Xeon Phi coprocessor SE10/7120 series (rev 20)
        Subsystem: Intel Corporation Device 7d99
        Flags: bus master, fast devsel, latency 0, IRQ 41, NUMA node 0
        Memory at 381c00000000 (64-bit, prefetchable) [size=16G]
        Memory at b5e00000 (64-bit, non-prefetchable) [size=128K]
        Capabilities: [44] Power Management version 3
        Capabilities: [4c] Express Endpoint, MSI 00
        Capabilities: [88] MSI: Enable- Count=1/16 Maskable- 64bit+
        Capabilities: [98] MSI-X: Enable+ Count=16 Masked-
        Capabilities: [100] Advanced Error Reporting
        Kernel driver in use: mic
```

cat /proc/vmallocinfo | grep '381c00000000'

```
0xffffc90040000000-0xffffc90440001000 17179873280 ioremap_work+0x24/0xd0 [mic] phys=381c00000000 ioremap
```

cat /proc/vmallocinfo | grep 'b5e00000'

```
0xffffc90006a80000-0xffffc90006aa1000  135168 adapter_init_device+0x202/0x4b0 [mic] phys=b5e00000 ioremap
```

■ Adding the MIC to the Host Network Services

Configure your system network layers so that the Xeon Phi MIC devices can be daisy chained together and be part of your system. The MIC card is a high performance computer by itself that can be booted, configured, and made available for offloading math intensive computations. To have the MIC configured on your system, create a bridge to combine the multiple network adapters: your MIC cards and the different adapters can then talk to each others as if directly connected to a normal network switch.

In the directory /etc/sysconfig/network-scripts edit the file ifcfg-br0:

```
DEVICE=br0
TYPE=Bridge
ONBOOT=yes
DELAY=0
NM_CONTROLLED="no"
BOOTPROTO=static
IPADDR=192.168.0.110
NETMASK=255.255.255.0
GATEWAY=192.168.0.1
DNS1=209.18.47.61
MTU=64512
```

Find all your network devices and have them conneted to that bridge: Issue the command:

ifconfig

you will see which devices are available, then for each device have an entry in the /etc/sysconfig/network-scripts.

For example, if ifconfig reports eno1 and enp2s0, then edit both files:

```
ifcfg-eno1:
-----------
DEVICE=eno1
NAME=eno1
NM_CONTROLLED=no
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
BRIDGE=br0
```

The mic0 device is bridged to br0

vi /etc/sysconfig/network-scripts/ifcfg-mic0

```
ifcfg-mic0:
-----------
DEVICE=mic0
ONBOOT=yes
NM_CONTROLLED="no"
BRIDGE=br0
MTU=64512
```

```
MACADDR=0a:c9:28:ae:ba:4f
```

> The entry NM_CONTROLLED=no is important to tell the network manager not to poll and interfere with the configuration.

Each MIC has a configuration file, it is located in the /etc/mpss directory. We will configure the MIC0 Edit the /etc/mpss/mic0.cfg

■ Configuring the MIC for Bootstrapping

```
# vi /etc/mpss/mic0.conf

Version 1 1

# Include configuration common to all MIC cards
Include default.conf

# Include all additional functionality configuration files by default
Include "conf.d/*.conf"

# Base filesystem for embedded Linux file system
Base CPIO /usr/share/mpss/boot/initramfs-knightscorner.cpio.gz

# Family type of MIC card
Family x100

# MPSS version
MPSSVersion 3.x

# Root device for MIC card
RootDevice StaticRamFS /var/mpss/mic0.image.cpio.gz

# Unique per card files for embedded Linux file system
### MicDir /var/mpss/mic0 /var/mpss/mic0.filelist  # The /var/mpss/mic0.filelist filelist argument deprec
MicDir /var/mpss/mic0

# Hostname to assign to MIC card
HostName mic0.local

# MAC address configuration
MacAddrs 0a:c9:28:ae:ba:4f 8e:d9:75:b8:3e:60

Network class=StaticBridge bridge=br0 micip=192.168.0.31 modhost=yes modcard=yes

# MIC OS Verbose messages to console
VerboseLogging Enabled

# MIC OS image
OSimage /usr/share/mpss/boot/bzImage-knightscorner /usr/share/mpss/boot/System.map-knightscorner

# Boot MIC card when MPSS stack is started
BootOnStart Enabled

# Control card power state setting
PowerManagement cpufreq_on;corec6_off;pc3_on;pc6_off

Cgroup memory=disabled

Overlay Simple /pubovl/tools /tmp On
```

■ Configuring the MIC KnightsCorner for Bootstrapping

If your HPMC has the KnightsCorner MIC the following mic0.conf is used.

vi /etc/mpss/mic0.conf

```
# MIC OS image
OSimage /usr/share/mpss/boot/bzImage-knightscorner /usr/share/mpss/boot/System.map-knightscorner
```

■ Booting the MIC

If your HPMC computer has a Xeon Phi CPU 7250 processor then there is no need for MPSS as the cores are available like native CPU cores.

If your HPMC computer has a MIC processor Xeon Phi coprocessor 71xx then MPSS is required and a network drivers must be started from the kernel during kernel startup.

If your HPMC computer has a MIC processor Xeon Phi coprocessor 72xx then MPSS is required and the network card is started as a service after kernel startup.

■ Adding a Swap Space to the MIC

On the HPMC

mkdir /ramfs

mount -t ramfs ramfs /ramfs/

ls /ramfs

dd bs=512M if=/dev/zero of=/ramfs/ram1 count=48

ls -l /sys/class/mic/mic0/virtblk_file

cat /sys/class/mic/mic0/virtblk_file

echo /ramfs/ram1 > /sys/class/mic/mic0/virtblk_file

cat /sys/class/mic/mic0/virtblk_file

df -a | grep ramfs

To add the swap to the MIC configuration, edit the default.conf within the MIC and add the following two lines:

vim /etc/mpss/default.conf

Add the following two lines:

```
ExtraCommandLine "vfs_read_optimization=on"
ExtraCommandLine "vfs_write_optimization=on"
```

service mpss stop

micctrl --resetconfig

service mpss start

ssh to the MIC

ssh mic0

On the MIC

/sbin/modprobe mic_virtblk

/sbin/modprobe -l | grep virt

/sbin/mkswap /dev/vda

/sbin/swapon /dev/vda

cat /proc/swaps

free -m

Use ramfs, AS IT IS NOT POSSIBLE TO USE tmpfs because cannot hold loop mounts and ramfs has no bmap call

■ Adding Users to the MIC

Each HPMC computer has its own MIC with mic0 as a hostname. When you login to your HPMC, you can ssh to the mic0: ssh mic0.

mic0 is only accessible as a Linux system from the board itself, this means you need to set up ssh-keygen, then you need to bring the mic0 interface up.

Here we will add the user "elsa" to the MIC of hpmc7. First add the user to hpmc7

adduser elsa

passwd elsa

usermod -a -G hpcusr elsa
needed to access /dev/mic/*

gpasswd -d hpcusr elsa
to remove elsa from group hpcusr

ssh-keygen for user elsa, and copy the key from /home/elsa/.ssh:

cat /elsa/.ssh/id_rsa.pub > /var/mpss/mic0/elsa/.ssh/authorized_keys

If your MIC is up then you need to reset it:

micctrl --resetconfig -v

check MIC if is in ready state

micctrl -s

If MIC is not in ready state then reset it

micctrl -rw

check MIC if is in ready state

micctrl -s

If MIC cannot be in ready state then you need to flash it (micflash -update) and reboot the host, and reset the MIC (micctrl -rw) until it is in ready state (mictrl -s) once the MIC is in ready state then bring up MPSS:

systemctl start mpss

for physical condition and information about the MIC:

micinfo

micctrl --adduser=elsa
do it after creating ssh-keygen on host

This will copy the .ssh/id_rsa.pub and copy the same to .ssh/authorize d_keys on elsa's mic0 home, it is the same as on /var/mpss/mic0/home/elsa/.ssh/

Login as elsa and ssh to mic0:

su - elsa

ssh mic0
now user elsa can ssh to the MIC

■ MIC hostname

Each HPMC computer has its own MIC with mic0 as a hostname. When you login to your HPMC, you can ssh to the mic0: ssh mic0.

ssh mic0

The HPMC mic0 is also visible to any neighboring HPMCs on the network, however its hostname must be prefixed with the HPMC hostname. For example, the HPMC computer whose hostname is hpmc7 will have the MIC whose hostname is hpmc7mic0. Computer on the same network can ssh to hpmc7's MIC using this command:

ssh hpmc7mic0

Before you ssh to a MIC you need to make sure the mpss is up on the hostname where the MIC card is inserted.

systemctl status mpss
make sure mpss service is started

■ Mount NFS to the MIC

For example, the host mm03 is an HPMC computer that has a MIC. The following procedure shows how to mount the directory /mm03fs over NFS to the MIC.

On host mm03 have NFS started at boot:

systemctl start nfs-server

systemctl enable nfs-server
(enable to start after each reboot)

Add the NFS to the MIC mic0 using the following command

micctrl --addnfs=mm03:/mm03fs -d /mm03fs mic0 -vv

```
   [Info] mic0: [Parse] /etc/mpss/mic0.conf
   [Info] mic0: [Parse] Configuration version 1.1
   [Info] mic0: [Parse] /etc/mpss/default.conf
[Warning] mic0: Server mm03 may not be reachable if the internal bridge br0 is not routed out of the host
[Warning] Modified existing NFS entry for MIC card path '/mm03fs'
[Filesys] mic0: Update /var/mpss/mic0/etc/fstab remove NFS /mm03fs entry
[Filesys] mic0: Update /var/mpss/mic0/etc/fstab with NFS mount mm03:/mm03fs at /mm03fs
```

Reset the MIC

micctrl -r

Check that the reset is successful

dmesg

```
[54291.465126] mic0: Transition from state resetting to ready
 18 ...
 19 [54290.463181] mic0: Resetting (Post Code 10)
 20 [54291.465114] mic0: Resetting (Post Code 12)
 21 [54291.465126] mic0: Transition from state resetting to ready
```

Restart the MIC

```
# systemctl restart mpss
```

```
# micctrl --rootdev=RamFs --target=/etc/mpss/lg3.cpio.gz -vv

   [Info] mic0: [Parse] /etc/mpss/mic0.conf
   [Info] mic0: [Parse] Configuration version 1.1
   [Info] mic0: [Parse] /etc/mpss/default.conf
[Filesys] mic0: Update RootDevice in /etc/mpss/mic0.conf
```

■ Enable the MIC mpss Service at Boot

MPSS must be installed on the host computer. The host computer is the system where the MIC card has been installed. Make sure MPSS is installed on the HPMC. On the HPMC the following directories and configuration files are evidence that the MPSS has been installed on the HPMC:

```
/opt/mpss
/etc/mpss/mic0.conf
/var/mpss/mic0
```

To enable MPSS service so that the MPSS is up after each reboot. Assuming you won't make changes to the MIC kernel image:

```
# systemctl enable mpss
```

The MIC kernel image can be customized, see **HOWTO-BUILD-MIC-KERNEL-IMAGE**
ref:
/var/mpss/mic0.image.*.cpio.gz
/etc/mpss/*.cpio.gz

Note that your HPMC MIC linux kernel has already been built and
contains many packages like perl, sh, perf, pstree, etc.

2-2 MPI Programming Offloading to MIC

This chapter shows how to use your HPMC calculator to write openMP (OMP) programs to be run on the MIC. The programs are being offloaded to the MIC Co-Processor.

■ OFFLOADING TO MIC WITH Direct Access Programming Library

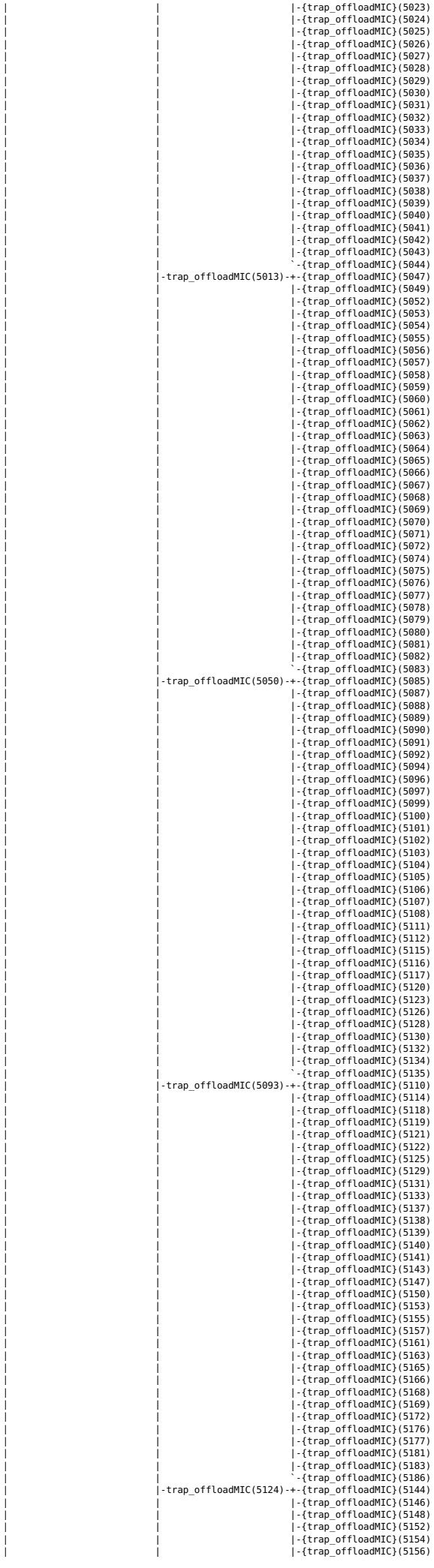
shm, shared memory transport (used for intra-node communication only). dapl, direct access programming library,

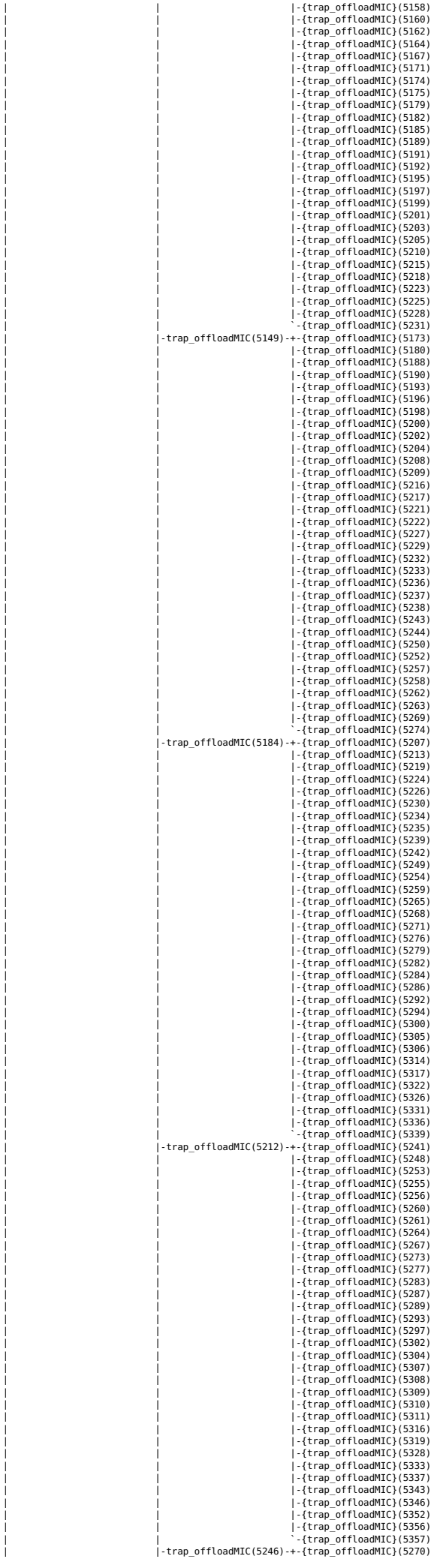
```
# OMP_NUM_THREADS=32 I_MPI_DAPL_PROVIDER_LIST=ofa-v2-mlx4_0-1u I_MPI_PERHOST=1
I_MPI_DEBUG=5 I_MPI_FABRICS=shm:dapl I_MPI_FALLBACK=0 /opt/INTEL-XE-2017-
update7/compilers_and_libraries_2017.7.259/linux/mpi/intel64/bin/mpiexec.hydra -n 10 -hosts hpmc9
./trap_offload
```

```
02:36 root@HPMC9: /mm03fs/MPI #  OMP_NUM_THREADS=32 I_MPI_DAPL_PROVIDER_LIST=ofa-v2-mlx4_0-1u I_MPI_PERHOST=1 I_MPI_DEBUG=5 I_MPI_FAI
[0] MPI startup(): Multi-threaded optimized library
[3] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[2] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[4] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[1] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[5] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[7] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[8] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[6] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[9] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[0] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[4] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[4] MPI startup(): shm and dapl data transfer modes
[2] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[3] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[2] MPI startup(): shm and dapl data transfer modes
[3] MPI startup(): shm and dapl data transfer modes
[1] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[1] MPI startup(): shm and dapl data transfer modes
[7] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[8] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[8] MPI startup(): shm and dapl data transfer modes
[7] MPI startup(): shm and dapl data transfer modes
[6] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[9] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[5] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[6] MPI startup(): shm and dapl data transfer modes
[9] MPI startup(): shm and dapl data transfer modes
[0] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[5] MPI startup(): shm and dapl data transfer modes
[0] MPI startup(): shm and dapl data transfer modes
[0] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[0] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[1] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[1] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[2] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[2] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[3] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[3] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[4] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[4] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[5] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[5] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[6] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[6] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[7] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[7] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[8] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[8] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[9] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[9] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[0] MPI startup(): Rank    Pid      Node name  Pin cpu
[0] MPI startup(): 0       42442    HPMC9      {0,1,2,3,4,5,56,57,58,59,60}
[0] MPI startup(): 1       42443    HPMC9      {6,7,8,9,10,61,62,63,64,65,66}
[0] MPI startup(): 2       42444    HPMC9      {11,12,13,14,15,16,67,68,69,70,71}
[0] MPI startup(): 3       42445    HPMC9      {17,18,19,20,21,72,73,74,75,76,77}
[0] MPI startup(): 4       42446    HPMC9      {22,23,24,25,26,27,78,79,80,81,82}
[0] MPI startup(): 5       42447    HPMC9      {28,29,30,31,32,83,84,85,86,87,88}
[0] MPI startup(): 6       42448    HPMC9      {33,34,35,36,37,38,89,90,91,92,93}
[0] MPI startup(): 7       42449    HPMC9      {39,40,41,42,43,94,95,96,97,98,99}
[0] MPI startup(): 8       42450    HPMC9      {44,45,46,47,48,49,100,101,102,103,104}
[0] MPI startup(): 9       42451    HPMC9      {50,51,52,53,54,105,106,107,108,109,110}
[0] MPI startup(): I_MPI_DAPL_PROVIDER_LIST=ofa-v2-mlx4_0-1u
[0] MPI startup(): I_MPI_DEBUG=5
[0] MPI startup(): I_MPI_FABRICS=shm:dapl
[0] MPI startup(): I_MPI_FALLBACK=0
[0] MPI startup(): I_MPI_INFO_NUMA_NODE_MAP=i40iw0:0,i40iw1:0,qib0:0,qib1:0,mic0:1
[0] MPI startup(): I_MPI_INFO_NUMA_NODE_NUM=2
[0] MPI startup(): I_MPI_PIN_MAPPING=10:0 0,1 6,2 11,3 17,4 22,5 28,6 33,7 39,8 44,9 50
rank 0 of 10 on HPMC9: 40.471704 seconds
integral = 1.856399, time = 45.410044
rank 1 of 10 on HPMC9: 41.719513 seconds
rank 2 of 10 on HPMC9: 43.035928 seconds
rank 3 of 10 on HPMC9: 44.055897 seconds
rank 4 of 10 on HPMC9: 44.575862 seconds
rank 5 of 10 on HPMC9: 45.216055 seconds
rank 6 of 10 on HPMC9: 45.299586 seconds
rank 7 of 10 on HPMC9: 44.915891 seconds
rank 8 of 10 on HPMC9: 45.410034 seconds
rank 9 of 10 on HPMC9: 45.268067 seconds
```

On the MIC we can reveal what is being run by the offloaded program.

```
(HPMC9) 01:32 root@mic0: ~ #  pstree -up
init(1)-+-coi_daemon(4354,micuser)-+-trap_offloadMIC(5004)-+-{trap_offloadMIC}(5009)
        |                          |                        |-{trap_offloadMIC}(5011)
        |                          |                        |-{trap_offloadMIC}(5012)
        |                          |                        |-{trap_offloadMIC}(5014)
        |                          |                        |-{trap_offloadMIC}(5015)
        |                          |                        |-{trap_offloadMIC}(5017)
        |                          |                        |-{trap_offloadMIC}(5018)
        |                          |                        |-{trap_offloadMIC}(5019)
        |                          |                        |-{trap_offloadMIC}(5020)
        |                          |                        |-{trap_offloadMIC}(5021)
        |                          |                        |-{trap_offloadMIC}(5022)
```


|-{trap_offloadMIC}(5023)
|-{trap_offloadMIC}(5024)
|-{trap_offloadMIC}(5025)
|-{trap_offloadMIC}(5026)
|-{trap_offloadMIC}(5027)
|-{trap_offloadMIC}(5028)
|-{trap_offloadMIC}(5029)
|-{trap_offloadMIC}(5030)
|-{trap_offloadMIC}(5031)
|-{trap_offloadMIC}(5032)
|-{trap_offloadMIC}(5033)
|-{trap_offloadMIC}(5034)
|-{trap_offloadMIC}(5035)
|-{trap_offloadMIC}(5036)
|-{trap_offloadMIC}(5037)
|-{trap_offloadMIC}(5038)
|-{trap_offloadMIC}(5039)
|-{trap_offloadMIC}(5040)
|-{trap_offloadMIC}(5041)
|-{trap_offloadMIC}(5042)
|-{trap_offloadMIC}(5043)
`-{trap_offloadMIC}(5044)
|-trap_offloadMIC(5013)-+-{trap_offloadMIC}(5047)
|-{trap_offloadMIC}(5049)
|-{trap_offloadMIC}(5052)
|-{trap_offloadMIC}(5053)
|-{trap_offloadMIC}(5054)
|-{trap_offloadMIC}(5055)
|-{trap_offloadMIC}(5056)
|-{trap_offloadMIC}(5057)
|-{trap_offloadMIC}(5058)
|-{trap_offloadMIC}(5059)
|-{trap_offloadMIC}(5060)
|-{trap_offloadMIC}(5061)
|-{trap_offloadMIC}(5062)
|-{trap_offloadMIC}(5063)
|-{trap_offloadMIC}(5064)
|-{trap_offloadMIC}(5065)
|-{trap_offloadMIC}(5066)
|-{trap_offloadMIC}(5067)
|-{trap_offloadMIC}(5068)
|-{trap_offloadMIC}(5069)
|-{trap_offloadMIC}(5070)
|-{trap_offloadMIC}(5071)
|-{trap_offloadMIC}(5072)
|-{trap_offloadMIC}(5074)
|-{trap_offloadMIC}(5075)
|-{trap_offloadMIC}(5076)
|-{trap_offloadMIC}(5077)
|-{trap_offloadMIC}(5078)
|-{trap_offloadMIC}(5079)
|-{trap_offloadMIC}(5080)
|-{trap_offloadMIC}(5081)
|-{trap_offloadMIC}(5082)
`-{trap_offloadMIC}(5083)
|-trap_offloadMIC(5050)-+-{trap_offloadMIC}(5085)
|-{trap_offloadMIC}(5087)
|-{trap_offloadMIC}(5088)
|-{trap_offloadMIC}(5089)
|-{trap_offloadMIC}(5090)
|-{trap_offloadMIC}(5091)
|-{trap_offloadMIC}(5092)
|-{trap_offloadMIC}(5094)
|-{trap_offloadMIC}(5096)
|-{trap_offloadMIC}(5097)
|-{trap_offloadMIC}(5099)
|-{trap_offloadMIC}(5100)
|-{trap_offloadMIC}(5101)
|-{trap_offloadMIC}(5102)
|-{trap_offloadMIC}(5103)
|-{trap_offloadMIC}(5104)
|-{trap_offloadMIC}(5105)
|-{trap_offloadMIC}(5106)
|-{trap_offloadMIC}(5107)
|-{trap_offloadMIC}(5108)
|-{trap_offloadMIC}(5111)
|-{trap_offloadMIC}(5112)
|-{trap_offloadMIC}(5115)
|-{trap_offloadMIC}(5116)
|-{trap_offloadMIC}(5117)
|-{trap_offloadMIC}(5120)
|-{trap_offloadMIC}(5123)
|-{trap_offloadMIC}(5126)
|-{trap_offloadMIC}(5128)
|-{trap_offloadMIC}(5130)
|-{trap_offloadMIC}(5132)
|-{trap_offloadMIC}(5134)
`-{trap_offloadMIC}(5135)
|-trap_offloadMIC(5093)-+-{trap_offloadMIC}(5110)
|-{trap_offloadMIC}(5114)
|-{trap_offloadMIC}(5118)
|-{trap_offloadMIC}(5119)
|-{trap_offloadMIC}(5121)
|-{trap_offloadMIC}(5122)
|-{trap_offloadMIC}(5125)
|-{trap_offloadMIC}(5129)
|-{trap_offloadMIC}(5131)
|-{trap_offloadMIC}(5133)
|-{trap_offloadMIC}(5137)
|-{trap_offloadMIC}(5138)
|-{trap_offloadMIC}(5139)
|-{trap_offloadMIC}(5140)
|-{trap_offloadMIC}(5141)
|-{trap_offloadMIC}(5143)
|-{trap_offloadMIC}(5147)
|-{trap_offloadMIC}(5150)
|-{trap_offloadMIC}(5153)
|-{trap_offloadMIC}(5155)
|-{trap_offloadMIC}(5157)
|-{trap_offloadMIC}(5161)
|-{trap_offloadMIC}(5163)
|-{trap_offloadMIC}(5165)
|-{trap_offloadMIC}(5166)
|-{trap_offloadMIC}(5168)
|-{trap_offloadMIC}(5169)
|-{trap_offloadMIC}(5172)
|-{trap_offloadMIC}(5176)
|-{trap_offloadMIC}(5177)
|-{trap_offloadMIC}(5181)
|-{trap_offloadMIC}(5183)
`-{trap_offloadMIC}(5186)
|-trap_offloadMIC(5124)-+-{trap_offloadMIC}(5144)
|-{trap_offloadMIC}(5146)
|-{trap_offloadMIC}(5148)
|-{trap_offloadMIC}(5152)
|-{trap_offloadMIC}(5154)
|-{trap_offloadMIC}(5156)


|-{trap_offloadMIC}(5158)
|-{trap_offloadMIC}(5160)
|-{trap_offloadMIC}(5162)
|-{trap_offloadMIC}(5164)
|-{trap_offloadMIC}(5167)
|-{trap_offloadMIC}(5171)
|-{trap_offloadMIC}(5174)
|-{trap_offloadMIC}(5175)
|-{trap_offloadMIC}(5179)
|-{trap_offloadMIC}(5182)
|-{trap_offloadMIC}(5185)
|-{trap_offloadMIC}(5189)
|-{trap_offloadMIC}(5191)
|-{trap_offloadMIC}(5192)
|-{trap_offloadMIC}(5195)
|-{trap_offloadMIC}(5197)
|-{trap_offloadMIC}(5199)
|-{trap_offloadMIC}(5201)
|-{trap_offloadMIC}(5203)
|-{trap_offloadMIC}(5205)
|-{trap_offloadMIC}(5210)
|-{trap_offloadMIC}(5215)
|-{trap_offloadMIC}(5218)
|-{trap_offloadMIC}(5223)
|-{trap_offloadMIC}(5225)
|-{trap_offloadMIC}(5228)
`-{trap_offloadMIC}(5231)
|-trap_offloadMIC(5149)-+-{trap_offloadMIC}(5173)
|-{trap_offloadMIC}(5180)
|-{trap_offloadMIC}(5188)
|-{trap_offloadMIC}(5190)
|-{trap_offloadMIC}(5193)
|-{trap_offloadMIC}(5196)
|-{trap_offloadMIC}(5198)
|-{trap_offloadMIC}(5200)
|-{trap_offloadMIC}(5202)
|-{trap_offloadMIC}(5204)
|-{trap_offloadMIC}(5208)
|-{trap_offloadMIC}(5209)
|-{trap_offloadMIC}(5216)
|-{trap_offloadMIC}(5217)
|-{trap_offloadMIC}(5221)
|-{trap_offloadMIC}(5222)
|-{trap_offloadMIC}(5227)
|-{trap_offloadMIC}(5229)
|-{trap_offloadMIC}(5232)
|-{trap_offloadMIC}(5233)
|-{trap_offloadMIC}(5236)
|-{trap_offloadMIC}(5237)
|-{trap_offloadMIC}(5238)
|-{trap_offloadMIC}(5243)
|-{trap_offloadMIC}(5244)
|-{trap_offloadMIC}(5250)
|-{trap_offloadMIC}(5252)
|-{trap_offloadMIC}(5257)
|-{trap_offloadMIC}(5258)
|-{trap_offloadMIC}(5262)
|-{trap_offloadMIC}(5263)
|-{trap_offloadMIC}(5269)
`-{trap_offloadMIC}(5274)
|-trap_offloadMIC(5184)-+-{trap_offloadMIC}(5207)
|-{trap_offloadMIC}(5213)
|-{trap_offloadMIC}(5219)
|-{trap_offloadMIC}(5224)
|-{trap_offloadMIC}(5226)
|-{trap_offloadMIC}(5230)
|-{trap_offloadMIC}(5234)
|-{trap_offloadMIC}(5235)
|-{trap_offloadMIC}(5239)
|-{trap_offloadMIC}(5242)
|-{trap_offloadMIC}(5249)
|-{trap_offloadMIC}(5254)
|-{trap_offloadMIC}(5259)
|-{trap_offloadMIC}(5265)
|-{trap_offloadMIC}(5268)
|-{trap_offloadMIC}(5271)
|-{trap_offloadMIC}(5276)
|-{trap_offloadMIC}(5279)
|-{trap_offloadMIC}(5282)
|-{trap_offloadMIC}(5284)
|-{trap_offloadMIC}(5286)
|-{trap_offloadMIC}(5292)
|-{trap_offloadMIC}(5294)
|-{trap_offloadMIC}(5300)
|-{trap_offloadMIC}(5305)
|-{trap_offloadMIC}(5306)
|-{trap_offloadMIC}(5314)
|-{trap_offloadMIC}(5317)
|-{trap_offloadMIC}(5322)
|-{trap_offloadMIC}(5326)
|-{trap_offloadMIC}(5331)
|-{trap_offloadMIC}(5336)
`-{trap_offloadMIC}(5339)
|-trap_offloadMIC(5212)-+-{trap_offloadMIC}(5241)
|-{trap_offloadMIC}(5248)
|-{trap_offloadMIC}(5253)
|-{trap_offloadMIC}(5255)
|-{trap_offloadMIC}(5256)
|-{trap_offloadMIC}(5260)
|-{trap_offloadMIC}(5261)
|-{trap_offloadMIC}(5264)
|-{trap_offloadMIC}(5267)
|-{trap_offloadMIC}(5273)
|-{trap_offloadMIC}(5277)
|-{trap_offloadMIC}(5283)
|-{trap_offloadMIC}(5287)
|-{trap_offloadMIC}(5289)
|-{trap_offloadMIC}(5293)
|-{trap_offloadMIC}(5297)
|-{trap_offloadMIC}(5302)
|-{trap_offloadMIC}(5304)
|-{trap_offloadMIC}(5307)
|-{trap_offloadMIC}(5308)
|-{trap_offloadMIC}(5309)
|-{trap_offloadMIC}(5310)
|-{trap_offloadMIC}(5311)
|-{trap_offloadMIC}(5316)
|-{trap_offloadMIC}(5319)
|-{trap_offloadMIC}(5328)
|-{trap_offloadMIC}(5333)
|-{trap_offloadMIC}(5337)
|-{trap_offloadMIC}(5343)
|-{trap_offloadMIC}(5346)
|-{trap_offloadMIC}(5352)
|-{trap_offloadMIC}(5356)
`-{trap_offloadMIC}(5357)
|-trap_offloadMIC(5246)-+-{trap_offloadMIC}(5270)

```
|                         |                         |-{trap_offloadMIC}(5280)
|                         |                         |-{trap_offloadMIC}(5285)
|                         |                         |-{trap_offloadMIC}(5288)
|                         |                         |-{trap_offloadMIC}(5290)
|                         |                         |-{trap_offloadMIC}(5291)
|                         |                         |-{trap_offloadMIC}(5295)
|                         |                         |-{trap_offloadMIC}(5296)
|                         |                         |-{trap_offloadMIC}(5298)
|                         |                         |-{trap_offloadMIC}(5299)
|                         |                         |-{trap_offloadMIC}(5313)
|                         |                         |-{trap_offloadMIC}(5318)
|                         |                         |-{trap_offloadMIC}(5321)
|                         |                         |-{trap_offloadMIC}(5323)
|                         |                         |-{trap_offloadMIC}(5327)
|                         |                         |-{trap_offloadMIC}(5330)
|                         |                         |-{trap_offloadMIC}(5332)
|                         |                         |-{trap_offloadMIC}(5335)
|                         |                         |-{trap_offloadMIC}(5340)
|                         |                         |-{trap_offloadMIC}(5344)
|                         |                         |-{trap_offloadMIC}(5347)
|                         |                         |-{trap_offloadMIC}(5349)
|                         |                         |-{trap_offloadMIC}(5353)
|                         |                         |-{trap_offloadMIC}(5358)
|                         |                         |-{trap_offloadMIC}(5363)
|                         |                         |-{trap_offloadMIC}(5365)
|                         |                         |-{trap_offloadMIC}(5367)
|                         |                         |-{trap_offloadMIC}(5368)
|                         |                         |-{trap_offloadMIC}(5370)
|                         |                         |-{trap_offloadMIC}(5371)
|                         |                         |-{trap_offloadMIC}(5374)
|                         |                         |-{trap_offloadMIC}(5375)
|                         |                         `-{trap_offloadMIC}(5377)
|                         |-trap_offloadMIC(5272)-+-{trap_offloadMIC}(5303)
|                         |                         |-{trap_offloadMIC}(5315)
|                         |                         |-{trap_offloadMIC}(5320)
|                         |                         |-{trap_offloadMIC}(5324)
|                         |                         |-{trap_offloadMIC}(5325)
|                         |                         |-{trap_offloadMIC}(5329)
|                         |                         |-{trap_offloadMIC}(5334)
|                         |                         |-{trap_offloadMIC}(5338)
|                         |                         |-{trap_offloadMIC}(5341)
|                         |                         |-{trap_offloadMIC}(5342)
|                         |                         |-{trap_offloadMIC}(5345)
|                         |                         |-{trap_offloadMIC}(5348)
|                         |                         |-{trap_offloadMIC}(5350)
|                         |                         |-{trap_offloadMIC}(5351)
|                         |                         |-{trap_offloadMIC}(5354)
|                         |                         |-{trap_offloadMIC}(5355)
|                         |                         |-{trap_offloadMIC}(5359)
|                         |                         |-{trap_offloadMIC}(5360)
|                         |                         |-{trap_offloadMIC}(5361)
|                         |                         |-{trap_offloadMIC}(5362)
|                         |                         |-{trap_offloadMIC}(5364)
|                         |                         |-{trap_offloadMIC}(5366)
|                         |                         |-{trap_offloadMIC}(5369)
|                         |                         |-{trap_offloadMIC}(5372)
|                         |                         |-{trap_offloadMIC}(5373)
|                         |                         |-{trap_offloadMIC}(5376)
|                         |                         |-{trap_offloadMIC}(5378)
|                         |                         |-{trap_offloadMIC}(5379)
|                         |                         |-{trap_offloadMIC}(5380)
|                         |                         |-{trap_offloadMIC}(5381)
|                         |                         |-{trap_offloadMIC}(5382)
|                         |                         |-{trap_offloadMIC}(5383)
|                         |                         `-{trap_offloadMIC}(5384)
|                         |-{coi_daemon}(4356)
|                         `-{coi_daemon}(5006)
|-getty(4370)
|-klogd(4333)
|-mpssd(4339)-+-{mpssd}(4341)
|             `-{mpssd}(4342)
|-portmap(4290,1)
|-sshd(4322)---sshd(4372)---bash(4374)---pstree(5385)
|-syslogd(4330)
`-udevd(4113)-+-udevd(4336)
              `-udevd(4363)
```

■ MPI OFFLOADING TO TWO MICs

Testing with the mpiexec.hydra by offloading to two MICs on the hosts: hpmc9 and hpmc7.

```
# OMP_NUM_THREADS=32 I_MPI_DAPL_PROVIDER_LIST=ofa-v2-mlx4_0-1u I_MPI_PERHOST=1
I_MPI_DEBUG=5 I_MPI_FABRICS=shm:dapl I_MPI_FALLBACK=0 /opt/INTEL-XE-2017-
update7/compilers_and_libraries_2017.7.259/linux/mpi/intel64/bin/mpiexec.hydra -n 10 -hosts
hpmc9,hpmc7 ./trap_offload
```

```
[0] MPI startup(): Multi-threaded optimized library
[0] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[2] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[4] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[6] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[8] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[1] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[3] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[5] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[7] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[9] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[0] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[0] MPI startup(): shm and dapl data transfer modes
[2] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[2] MPI startup(): shm and dapl data transfer modes
[4] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[4] MPI startup(): shm and dapl data transfer modes
[8] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[6] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[6] MPI startup(): shm and dapl data transfer modes
[8] MPI startup(): shm and dapl data transfer modes
[1] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[3] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[5] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[7] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[1] MPI startup(): shm and dapl data transfer modes
[3] MPI startup(): shm and dapl data transfer modes
[5] MPI startup(): shm and dapl data transfer modes
[7] MPI startup(): shm and dapl data transfer modes
[9] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[9] MPI startup(): shm and dapl data transfer modes
```

```
[0] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[0] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[2] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[2] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[4] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[4] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[6] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[6] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[8] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[8] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[1] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[1] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[3] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[3] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[5] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[5] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[7] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[7] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[9] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[9] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[0] MPI startup(): Rank    Pid      Node name  Pin cpu
[0] MPI startup(): 0       42103    HPMC9      {0,1,2,3,4,5,6,7,8,9,10,56,57,58,59,60,61,62,63,64,65,66}
[0] MPI startup(): 1       32814    HPMC7      {0,1,2,3,4,5,6,7,8,9,10,11,12,13,68,69,70,71,72,73,74,75,76,77,78,79,80,81,136,13
                                    7,138,139,140,141,142,143,144,145,146,147,148,204,205,206,207,208,209,210,211,21
                                    2,213,214,215,216}
[0] MPI startup(): 2       42104    HPMC9      {11,12,13,14,15,16,17,18,19,20,21,67,68,69,70,71,72,73,74,75,76,77}
[0] MPI startup(): 3       32815    HPMC7      {14,15,16,17,18,19,20,21,22,23,24,25,26,82,83,84,85,86,87,88,89,90,91,92,93,94,14
                                    9,150,151,152,153,154,155,156,157,158,159,160,161,162,217,218,219,220,221,222,22
                                    3,224,225,226,227,228,229,230}
[0] MPI startup(): 4       42105    HPMC9      {22,23,24,25,26,27,28,29,30,31,32,78,79,80,81,82,83,84,85,86,87,88}
[0] MPI startup(): 5       32816    HPMC7      {27,28,29,30,31,32,33,34,35,36,37,38,39,40,95,96,97,98,99,100,101,102,103,104,105
                                    ,106,107,108,163,164,165,166,167,168,169,170,171,172,173,174,175,231,232,233,234
                                    ,235,236,237,238,239,240,241,242,243}
[0] MPI startup(): 6       42106    HPMC9      {33,34,35,36,37,38,39,40,41,42,43,89,90,91,92,93,94,95,96,97,98,99}
[0] MPI startup(): 7       32817    HPMC7      {41,42,43,44,45,46,47,48,49,50,51,52,53,109,110,111,112,113,114,115,116,117,118,1
                                    19,120,121,176,177,178,179,180,181,182,183,184,185,186,187,188,189,244,245,246,2
                                    47,248,249,250,251,252,253,254,255,256,257}
[0] MPI startup(): 8       42107    HPMC9      {44,45,46,47,48,49,50,51,52,53,54,100,101,102,103,104,105,106,107,108,109,110}
[0] MPI startup(): 9       32818    HPMC7      {54,55,56,57,58,59,60,61,62,63,64,65,66,67,122,123,124,125,126,127,128,129,130,13
                                    1,132,133,134,135,190,191,192,193,194,195,196,197,198,199,200,201,202,258,259,26
                                    0,261,262,263,264,265,266,267,268,269,270}
[0] MPI startup(): I_MPI_DAPL_PROVIDER_LIST=ofa-v2-mlx4_0-1u
[0] MPI startup(): I_MPI_DEBUG=5
[0] MPI startup(): I_MPI_FABRICS=shm:dapl
[0] MPI startup(): I_MPI_FALLBACK=0
[0] MPI startup(): I_MPI_INFO_NUMA_NODE_MAP=i40iw0:0,i40iw1:0,qib0:0,qib1:0,mic0:1
[0] MPI startup(): I_MPI_INFO_NUMA_NODE_NUM=2
[0] MPI startup(): I_MPI_PIN_MAPPING=5:0 0,2 11,4 22,6 33,8 44
rank 4 of 10 on HPMC9: 22.456818 seconds
rank 1 of 10 on HPMC7: 20.294824 seconds
rank 6 of 10 on HPMC9: 22.954011 seconds
rank 3 of 10 on HPMC7: 18.985285 seconds
rank 8 of 10 on HPMC9: 22.732662 seconds
rank 5 of 10 on HPMC7: 19.954661 seconds
rank 0 of 10 on HPMC9: 21.098834 seconds
integral = 1.856399, time = 22.954172
rank 7 of 10 on HPMC7: 19.414617 seconds
rank 2 of 10 on HPMC9: 21.718827 seconds
rank 9 of 10 on HPMC7: 20.453557 seconds
```

2-3 Programming and Running by MIC Native Loading

This chapter shows how to use your HPMC calculator to write OMP programs to be run on the MIC. The programs are being offloaded to the MIC Co-Processor.

■ Running the OMP Program on the MIC

The diffusion OMP program

-- Program Code 2.3.1 : [LISTING diffusion_omp.c] - [Diffusion OMP]

(raw text)

```
1.    #include <stdio.h>
2.    #include <stdlib.h>
3.    #include <string.h>
4.    #include <math.h>
5.    #include <time.h>
6.    #include <sys/time.h>
7.    #include <omp.h>
8.    #include <assert.h>
9.    #include <sys/mman.h>
10.
11.   #define REAL float
12.   #define NX (256)
13.   #define NXP nx
14.
15.   #ifndef M_PI
16.   #define M_PI (3.141592653589793238462 6)
17.   #endif
18.
19.
20.   void init(REAL *buff, const int nx, const int ny, const int nz,
21.             const REAL kx, const REAL ky, const REAL kz,
22.             const REAL dx, const REAL dy, const REAL dz,
23.             const REAL kappa, const REAL time) {
24.     REAL ax, ay, az;
25.     int jz, jy, jx;
26.     ax = exp(-kappa*time*(kx*kx));
27.     ay = exp(-kappa*time*(ky*ky));
28.     az = exp(-kappa*time*(kz*kz));
29.     for (jz = 0; jz < nz; jz++) {
30.       for (jy = 0; jy < ny; jy++) {
31.         for (jx = 0; jx < nx; jx++) {
32.           int j = jz*NXP*ny + jy*NXP + jx;
33.           REAL x = dx*((REAL)(jx + 0.5));
34.           REAL y = dy*((REAL)(jy + 0.5));
35.           REAL z = dz*((REAL)(jz + 0.5));
36.           REAL f0 = (REAL)0.125
37.             *(1.0 - ax*cos(kx*x))
38.             *(1.0 - ay*cos(ky*y))
39.             *(1.0 - az*cos(kz*z));
40.           buff[j] = f0;
41.         }
42.       }
43.     }
44.   }
45.
46.   REAL accuracy(const REAL *b1, REAL *b2, const int len) {
47.     REAL err = 0.0;
48.     int i;
49.     for (i = 0; i < len; i++) {
50.       err += (b1[i] - b2[i]) * (b1[i] - b2[i]);
51.     }
52.     return (REAL)sqrt(err/len);
53.   }
54.
55.   void diffusion_openmp(REAL *restrict f1, REAL *restrict f2, int nx, int
      ny, int nz,
56.                         REAL ce, REAL cw, REAL cn, REAL cs, REAL ct,
57.                         REAL cb, REAL cc, REAL dt, int count) {
58.   #pragma omp parallel
59.     {
60.       REAL *f1_t = f1;
```

```
61.         REAL *f2_t = f2;
62.
63.         for (int i = 0; i < count; ++i) {
64.     #pragma omp for collapse(2)
65.           for (int z = 0; z < nz; z++) {
66.             for (int y = 0; y < ny; y++) {
67.               for (int x = 0; x < nx; x++) {
68.                 int c, w, e, n, s, b, t;
69.                 c =  x + y * NXP + z * NXP * ny;
70.                 w = (x == 0)    ? c : c - 1;
71.                 e = (x == NXP-1) ? c : c + 1;
72.                 n = (y == 0)    ? c : c - NXP;
73.                 s = (y == ny-1) ? c : c + NXP;
74.                 b = (z == 0)    ? c : c - NXP * ny;
75.                 t = (z == nz-1) ? c : c + NXP * ny;
76.                 f2_t[c] = cc * f1_t[c] + cw * f1_t[w] + ce * f1_t[e]
77.                     + cs * f1_t[s] + cn * f1_t[n] + cb * f1_t[b] + ct *
      f1_t[t];
78.               }
79.             }
80.           }
81.           REAL *t = f1_t;
82.           f1_t = f2_t;
83.           f2_t = t;
84.         }
85.       }
86.       return;
87.     }
88.
89.
90.     static double cur_second(void) {
91.       struct timeval tv;
92.       gettimeofday(&tv, NULL);
93.       return (double)tv.tv_sec + (double)tv.tv_usec / 1000000.0;
94.     }
95.
96.
97.     void dump_result(REAL *f, int nx, int ny, int nz, char *out_path) {
98.       FILE *out = fopen(out_path, "w");
99.       assert(out);
100.      size_t nitems = nx * ny * nz;
101.      fwrite(f, sizeof(REAL), nitems, out);
102.      fclose(out);
103.    }
104.
105.    int main(int argc, char *argv[])
106.    {
107.
108.      struct timeval time_begin, time_end;
109.
110.      int    nx    = NX;
111.      int    ny    = NX;
112.      int    nz    = NX;
113.
114.      REAL *f1 = (REAL *)malloc(sizeof(REAL)*NX*NX*NX);
115.      REAL *f2 = (REAL *)malloc(sizeof(REAL)*NX*NX*NX);
116.      assert(f1 != MAP_FAILED);
117.      assert(f2 != MAP_FAILED);
118.      REAL *answer = (REAL *)malloc(sizeof(REAL) * NXP*ny*nz);
119.      REAL *f_final = NULL;
120.
121.      REAL   time  = 0.0;
122.      int    count = 0;
123.      int    nthreads;
124.
125.      REAL l, dx, dy, dz, kx, ky, kz, kappa, dt;
126.      REAL ce, cw, cn, cs, ct, cb, cc;
127.
128.      #pragma omp parallel
129.      #pragma omp master
130.        nthreads = omp_get_num_threads();
131.
132.      l = 1.0;
133.      kappa = 0.1;
134.      dx = dy = dz = l / nx;
135.      kx = ky = kz = 2.0 * M_PI;
136.      dt = 0.1*dx*dx / kappa;
137.      count = 0.1 / dt;
138.      f_final = (count % 2)? f2 : f1;
139.
140.      init(f1, nx, ny, nz, kx, ky, kz, dx, dy, dz, kappa, time);
141.
142.      ce = cw = kappa*dt/(dx*dx);
143.      cn = cs = kappa*dt/(dy*dy);
144.      ct = cb = kappa*dt/(dz*dz);
```

```
145.     cc = 1.0 - (ce + cw + cn + cs + ct + cb);
146.
147.     printf("Running diffusion kernel %d times\n", count); fflush(stdout);
148.     gettimeofday(&time_begin, NULL);
149.     diffusion_openmp(f1, f2, nx, ny, nz, ce, cw, cn, cs, ct, cb, cc,
150.                      dt, count);
151.     gettimeofday(&time_end, NULL);
152.     time = count * dt;
153.     dump_result(f_final, nx, ny, nz, "diffusion_result.dat");
154.
155.     init(answer, nx, ny, nz, kx, ky, kz, dx, dy, dz, kappa, time);
156.     REAL err = accuracy(f_final, answer, nx*ny*nz);
157.     double elapsed_time = (time_end.tv_sec - time_begin.tv_sec)
158.         + (time_end.tv_usec - time_begin.tv_usec)*1.0e-6;
159.     REAL mflops = (nx*ny*nz)*13.0*count/elapsed_time * 1.0e-06;
160.     double thput = (nx * ny * nz) * sizeof(REAL) * 3.0 * count
161.         / elapsed_time * 1.0e-09;
162.
163.     fprintf(stderr, "Elapsed time : %.3f (s)\n", elapsed_time);
164.     fprintf(stderr, "FLOPS        : %.3f (MFlops)\n", mflops);
165.     fprintf(stderr, "Throughput   : %.3f (GB/s)\n", thput);
166.     fprintf(stderr, "Accuracy     : %e\n", err);
167.
168.     free(f1);
169.     free(f2);
170.     return 0;
171.   }
```

HPMC 2022

On the host hpmc9 compile the program diffusion_omp.c

icc -qopenmp -O3 -std=c99 -mmic diffusion_omp.c -o diffusion_omp_xphi

Copy the compiled code to the MIC

scp diffusion_omp_xphi mic0:/root

Run the code on the MIC

(HPMC9) 00:42 root@mic0: ~ # ./diffusion_omp_xphi

```
(HPMC9) 00:42 root@mic0: ~ #  ./diffusion_omp_xphi
Running diffusion kernel 6553 times
Elapsed time : 129.454 (s)
FLOPS        : 11040.484 (MFlops)
Throughput   : 10.191 (GB/s)
Accuracy     : 1.363216e-05
```

On the MIC we can reveal the processes run by the program diffusion_omp_x.

(HPMC9) 00:41 root@mic0: ~ # pstree -up

```
(HPMC9) 00:41 root@mic0: ~ #  pstree -up
init(1)-+-coi_daemon(4354,micuser)---{coi_daemon}(4356)
        |-getty(4370)
        |-klogd(4333)
        |-mpssd(4339)-+-{mpssd}(4341)
        |             `-{mpssd}(4342)
        |-portmap(4290,1)
        |-sshd(4322)-+-sshd(4374)---bash(4376)---diffusion_omp_x(4387)-+-{diffusion_omp_x}(4388)
        |            |                                                 |-{diffusion_omp_x}(4389)
        |            |                                                 |-{diffusion_omp_x}(4390)
        |            |                                                 |-{diffusion_omp_x}(4391)
        |            |                                                 |-{diffusion_omp_x}(4392)
        |            |                                                 |-{diffusion_omp_x}(4393)
        |            |                                                 |-{diffusion_omp_x}(4394)
        |            |                                                 |-{diffusion_omp_x}(4395)
        |            |                                                 |-{diffusion_omp_x}(4396)
        |            |                                                 |-{diffusion_omp_x}(4397)
        |            |                                                 |-{diffusion_omp_x}(4398)
...
        |            |                                                 |-{diffusion_omp_x}(4604)
```

```
|           |                                                    |-{diffusion_omp_x}(4605)
|           |                                                    |-{diffusion_omp_x}(4606)
|           |                                                    |-{diffusion_omp_x}(4607)
|           |                                                    |-{diffusion_omp_x}(4608)
|           |                                                    |-{diffusion_omp_x}(4609)
|           |                                                    |-{diffusion_omp_x}(4610)
|           |                                                    |-{diffusion_omp_x}(4611)
|           |                                                    |-{diffusion_omp_x}(4612)
|           |                                                    |-{diffusion_omp_x}(4613)
|           |                                                    |-{diffusion_omp_x}(4614)
|           |                                                    `-{diffusion_omp_x}(4615)
|           `-sshd(4616)---bash(4618)---pstree(4621)
|-syslogd(4330)
`-udevd(4113)-+-udevd(4336)
              `-udevd(4363)
```

■ MIC Native Loading: micnativeloadex

The diffusion OMP program

-- Program Code 2.3.2 : [LISTING diffusion_base.c] - [Diffusion Program]

(raw text)

```
1.    #include <stdio.h>
2.    #include <stdlib.h>
3.    #include <string.h>
4.    #include <math.h>
5.    #include <time.h>
6.    #include <sys/time.h>
7.    #include <omp.h>
8.    #include <assert.h>
9.    #include <sys/mman.h>
10.
11.   #define REAL float
12.   #define NX (256)
13.   #define NXP nx
14.
15.   #ifndef M_PI
16.   #define M_PI (3.141592653589793238462 6)
17.   #endif
18.
19.
20.   void init(REAL *buff, const int nx, const int ny, const int nz,
21.             const REAL kx, const REAL ky, const REAL kz,
22.             const REAL dx, const REAL dy, const REAL dz,
23.             const REAL kappa, const REAL time) {
24.     REAL ax, ay, az;
25.     int jz, jy, jx;
26.     ax = exp(-kappa*time*(kx*kx));
27.     ay = exp(-kappa*time*(ky*ky));
28.     az = exp(-kappa*time*(kz*kz));
29.     for (jz = 0; jz < nz; jz++) {
30.       for (jy = 0; jy < ny; jy++) {
31.         for (jx = 0; jx < nx; jx++) {
32.           int j = jz*NXP*ny + jy*NXP + jx;
33.           REAL x = dx*((REAL)(jx + 0.5));
34.           REAL y = dy*((REAL)(jy + 0.5));
35.           REAL z = dz*((REAL)(jz + 0.5));
36.           REAL f0 = (REAL)0.125
37.             *(1.0 - ax*cos(kx*x))
38.             *(1.0 - ay*cos(ky*y))
39.             *(1.0 - az*cos(kz*z));
40.           buff[j] = f0;
41.         }
42.       }
43.     }
44.   }
45.
46.   REAL accuracy(const REAL *b1, REAL *b2, const int len) {
47.     REAL err = 0.0;
48.     int i;
49.     for (i = 0; i < len; i++) {
50.       err += (b1[i] - b2[i]) * (b1[i] - b2[i]);
51.     }
52.     return (REAL)sqrt(err/len);
53.   }
54.
55.   void diffusion_baseline(REAL *f1, REAL *f2, int nx, int ny, int nz,
```

```
56.                       REAL ce, REAL cw, REAL cn, REAL cs, REAL ct,
57.                       REAL cb, REAL cc, REAL dt,
58.                       int count) {
59.      int i;
60.      for (i = 0; i < count; ++i) {
61.        for (int z = 0; z < nz; z++) {
62.          for (int y = 0; y < ny; y++) {
63.            for (int x = 0; x < nx; x++) {
64.              int c, w, e, n, s, b, t;
65.              c =  x + y * NXP + z * NXP * ny;
66.              w = (x == 0)    ? c : c - 1;
67.              e = (x == NXP-1) ? c : c + 1;
68.              n = (y == 0)    ? c : c - NXP;
69.              s = (y == ny-1) ? c : c + NXP;
70.              b = (z == 0)    ? c : c - NXP * ny;
71.              t = (z == nz-1) ? c : c + NXP * ny;
72.              f2[c] = cc * f1[c] + cw * f1[w] + ce * f1[e]
73.                  + cs * f1[s] + cn * f1[n] + cb * f1[b] + ct * f1[t];
74.            }
75.          }
76.        }
77.        REAL *t = f1;
78.        f1 = f2;
79.        f2 = t;
80.      }
81.      return;
82.    }
83.
84.    static double cur_second(void) {
85.      struct timeval tv;
86.      gettimeofday(&tv, NULL);
87.      return (double)tv.tv_sec + (double)tv.tv_usec / 1000000.0;
88.    }
89.
90.
91.    void dump_result(REAL *f, int nx, int ny, int nz, char *out_path) {
92.      FILE *out = fopen(out_path, "w");
93.      assert(out);
94.      size_t nitems = nx * ny * nz;
95.      fwrite(f, sizeof(REAL), nitems, out);
96.      fclose(out);
97.    }
98.
99.    int main(int argc, char *argv[])
100.   {
101.
102.     struct timeval time_begin, time_end;
103.
104.     int    nx    = NX;
105.     int    ny    = NX;
106.     int    nz    = NX;
107.
108.     REAL *f1 = (REAL *)malloc(sizeof(REAL)*NX*NX*NX);
109.     REAL *f2 = (REAL *)malloc(sizeof(REAL)*NX*NX*NX);
110.     assert(f1 != MAP_FAILED);
111.     assert(f2 != MAP_FAILED);
112.     REAL *answer = (REAL *)malloc(sizeof(REAL) * NXP*ny*nz);
113.     REAL *f_final = NULL;
114.
115.     REAL   time  = 0.0;
116.     int    count = 0;
117.     int    nthreads;
118.
119.     REAL l, dx, dy, dz, kx, ky, kz, kappa, dt;
120.     REAL ce, cw, cn, cs, ct, cb, cc;
121.
122.     #pragma omp parallel
123.     #pragma omp master
124.       nthreads = omp_get_num_threads();
125.
126.     l = 1.0;
127.     kappa = 0.1;
128.     dx = dy = dz = l / nx;
129.     kx = ky = kz = 2.0 * M_PI;
130.     dt = 0.1*dx*dx / kappa;
131.     count = 0.1 / dt;
132.     f_final = (count % 2)? f2 : f1;
133.
134.     init(f1, nx, ny, nz, kx, ky, kz, dx, dy, dz, kappa, time);
135.
136.     ce = cw = kappa*dt/(dx*dx);
137.     cn = cs = kappa*dt/(dy*dy);
138.     ct = cb = kappa*dt/(dz*dz);
139.     cc = 1.0 - (ce + cw + cn + cs + ct + cb);
140.
```

```
141.     printf("Running diffusion kernel %d times\n", count); fflush(stdout);
142.     gettimeofday(&time_begin, NULL);
143.     diffusion_baseline(f1, f2, nx, ny, nz, ce, cw, cn, cs, ct, cb, cc,
144.                     dt, count);
145.     gettimeofday(&time_end, NULL);
146.     time = count * dt;
147.     dump_result(f_final, nx, ny, nz, "diffusion_result.dat");
148.
149.     init(answer, nx, ny, nz, kx, ky, kz, dx, dy, dz, kappa, time);
150.     REAL err = accuracy(f_final, answer, nx*ny*nz);
151.     double elapsed_time = (time_end.tv_sec - time_begin.tv_sec)
152.         + (time_end.tv_usec - time_begin.tv_usec)*1.0e-6;
153.     REAL mflops = (nx*ny*nz)*13.0*count/elapsed_time * 1.0e-06;
154.     double thput = (nx * ny * nz) * sizeof(REAL) * 3.0 * count
155.         / elapsed_time * 1.0e-09;
156.
157.     fprintf(stderr, "Elapsed time : %.3f (s)\n", elapsed_time);
158.     fprintf(stderr, "FLOPS        : %.3f (MFlops)\n", mflops);
159.     fprintf(stderr, "Throughput   : %.3f (GB/s)\n", thput);
160.     fprintf(stderr, "Accuracy     : %e\n", err);
161.
162.     free(f1);
163.     free(f2);
164.     return 0;
165.   }
```


On the host hpmc9 compile the diffusion_base.c

icc -qopenmp -O3 -std=c99 -mmic diffusion_base.c -o diffusion_base_xphi

Run the code on the host hpmc9

00:25 root@HPMC9: micnativeloadex diffusion_base_xphi -d mic0 -e "SINK_LD_LIBRARY_PATH=/miclib/"

```
00:25 root@HPMC9: ~/Code #  micnativeloadex diffusion_base_xphi -d mic0 -e "SINK_LD_LIBRARY_PATH=/micli
Running diffusion kernel 409 times
Elapsed time : 5.293 (s)
FLOPS        : 263.316 (MFlops)
Throughput   : 0.243 (GB/s)
Accuracy     : 1.749954e-05
```

On the MIC we can reveal the processes run by the program diffusion_base.

```
(HPMC9) 02:39 root@mic0: ~ #  pstree -up
init(1)-+-coi_daemon(4354,micuser)-+-diffusion_base_(6024)-+-{diffusion_base_}(6030)
        |                          |                        |-{diffusion_base_}(6031)
        |                          |                        |-{diffusion_base_}(6032)
        |                          |                        |-{diffusion_base_}(6033)
        |                          |                        |-{diffusion_base_}(6034)
        |                          |                        |-{diffusion_base_}(6035)
        |                          |                        |-{diffusion_base_}(6036)
        |                          |                        |-{diffusion_base_}(6037)
        |                          |                        |-{diffusion_base_}(6038)
...
        |                          |                        |-{diffusion_base_}(6245)
        |                          |                        |-{diffusion_base_}(6246)
        |                          |                        |-{diffusion_base_}(6247)
        |                          |                        |-{diffusion_base_}(6248)
        |                          |                        |-{diffusion_base_}(6249)
        |                          |                        |-{diffusion_base_}(6250)
        |                          |                        |-{diffusion_base_}(6251)
        |                          |                        |-{diffusion_base_}(6252)
        |                          |                        `-{diffusion_base_}(6253)
        |                          |-{coi_daemon}(4356)
        |                          `-{coi_daemon}(6026)
        |-getty(4370)
        |-klogd(4333)
        |-mpssd(4339)-+-{mpssd}(4341)
        |             `-{mpssd}(4342)
        |-portmap(4290,1)
        |-sshd(4322)-+-sshd(4374)---bash(4376)
```

```
|            `-sshd(4616)---bash(4618)---pstree(6254)
|-syslogd(4330)
`-udevd(4113)-+-udevd(4336)
              `-udevd(4363)
```

2-4 Matrix Multiplication Offloading to MIC

This chapter shows how to use your HPMC calculator to multiply matrices using openMP. The operations will be offloaded to the Intel Xeon Phi MIC.

Consider the program AxB.c shown in the following listing:

-- Program Code 2.4.1 : [LISTING AxB.c] - [Matrix Multiplication using openMP]

(raw text)

```
1.     #ifndef MIC_DEV
2.     #define MIC_DEV 0
3.     #endif
4.
5.     #include <stdio.h>
6.     #include <stdlib.h>
7.     #include <omp.h>
8.     #include <math.h>
9.     #include <sys/time.h>
10.
11.    double getTime(void) {
12.    struct timeval t;
13.    gettimeofday(&t,0);
14.    return ((double)t.tv_sec + ((double)t.tv_usec / 1000000.0));
15.    }
16.
17.    // ################################
18.    // openMP multiply matrices
19.    void doMult(int size, float (* restrict A)[size],
20.         float (* restrict B)[size], float (* restrict C)[size]
21.        ) {
22.
23.    #pragma offload target(mic:MIC_DEV) \
24.        in(A:length(size*size)) in( B:length(size*size))    \
25.        out(C:length(size*size))
26.        {
27.    #pragma omp parallel for default(none) shared(C,size)
28.            for (int i = 0; i < size; ++i)
29.                for (int j = 0; j < size; ++j)
30.                    C[i][j] =0.f;
31.            // matrix multiplication.
32.    #pragma omp parallel for default(none) shared(A,B,C,size)
33.            for (int i = 0; i < size; ++i)
34.                for (int k = 0; k < size; ++k)
35.                    for (int j = 0; j < size; ++j)
36.                      C[i][j] += A[i][k] * B[k][j];
37.        }
38.    }
39.
40.    // ################################
41.    int main(int argc, char *argv[]) {
42.        if(argc != 4) {
43.            fprintf(stderr,"Use: %s size nThreads iter\n",argv[0]);
44.            return -1;
45.        }
46.        int i,j,k;
47.        int size=atoi(argv[1]);
48.        int nThreads=atoi(argv[2]);
49.        int iter=atoi(argv[3]);
50.        omp_set_num_threads(nThreads);
51.        float (*restrict A)[size] = malloc(sizeof(float)*size*size);
52.        float (*restrict B)[size] = malloc(sizeof(float)*size*size);
53.        float (*restrict C)[size] = malloc(sizeof(float)*size*size);
54.        // init natrices A and B
55.    #pragma omp parallel for default(none) shared(A,B,size) private(i,j,k)
56.        for (i = 0; i < size; ++i) {
57.            for (j = 0; j < size; ++j) {
58.                A[i][j] = (float)((rand() % 3)-1);
59.                B[i][j] = (float)((rand() % 3)-1);
60.            }
61.        }
62.        // warmup
63.        doMult(size, A,B,C);
64.        double startTime = getTime();
65.        for (int i=0; i < iter; i++) {
66.            doMult(size, A,B,C);
67.        }
68.        double endTime = getTime();
69.        double lapsed = endTime-startTime;
70.        lapsed /= iter;
71.
```

```
72.        printf("%s nThrds %d size %d elapsedtime %g GFlop/s %g\n",
73.             argv[0], nThreads, size, lapsed, 2e-9*size*size*size/lapsed);
74.
75.        free(A); free(B); free(C);
76.        return 0;
77.    }
78.
```


Compile AxB.c to be executed on the MIC:

icc -O3 -mmic -qopenmp -L /opt/intel/lib/mic -Wno-unknown-pragmas -std=c99 -liomp5 AxB.c -o AxB.mic

The following script will run the pogram in a loop, incrementing the number of threads by 24 in each iteration up to a maximum of 340 threads.

```
#!/bin/ksh
export KMP_AFFINITY="granularity=thread,balanced"
i=24
while [ $i -lt 340 ]
do
   xphi AxB.mic -a "1000 $i 10"
   let i+=24;
done
```

The coi (coprocessor offload infrastructure) processes as the program is executing.

03:30 root@HPMC9: /mm03fs/DOBB # ./AxB.mic.sh

```
03:30 root@HPMC9: /mm03fs/DOBB #  ./AxB.mic.sh
/tmp/coi_procs/1/23650/AxB.mic nThrds 24 size 1000 elapsedtime 0.0400838 GFlop/s 49.8955
/tmp/coi_procs/1/23685/AxB.mic nThrds 48 size 1000 elapsedtime 0.0269411 GFlop/s 74.236
/tmp/coi_procs/1/23743/AxB.mic nThrds 72 size 1000 elapsedtime 0.027996 GFlop/s 71.4388
/tmp/coi_procs/1/23825/AxB.mic nThrds 96 size 1000 elapsedtime 0.0311322 GFlop/s 64.2422
/tmp/coi_procs/1/23931/AxB.mic nThrds 120 size 1000 elapsedtime 0.0186215 GFlop/s 107.403
/tmp/coi_procs/1/24060/AxB.mic nThrds 144 size 1000 elapsedtime 0.0262353 GFlop/s 76.2332
/tmp/coi_procs/1/24215/AxB.mic nThrds 168 size 1000 elapsedtime 0.0202382 GFlop/s 98.823
/tmp/coi_procs/1/24393/AxB.mic nThrds 192 size 1000 elapsedtime 0.0206483 GFlop/s 96.8603
/tmp/coi_procs/1/24595/AxB.mic nThrds 216 size 1000 elapsedtime 0.0144107 GFlop/s 138.786
/tmp/coi_procs/1/24822/AxB.mic nThrds 240 size 1000 elapsedtime 0.0197546 GFlop/s 101.242
/tmp/coi_procs/1/25071/AxB.mic nThrds 264 size 1000 elapsedtime 0.0258059 GFlop/s 77.5017
/tmp/coi_procs/1/25345/AxB.mic nThrds 288 size 1000 elapsedtime 0.0331669 GFlop/s 60.3011
/tmp/coi_procs/1/25643/AxB.mic nThrds 312 size 1000 elapsedtime 0.0454103 GFlop/s 44.0429
/tmp/coi_procs/1/25965/AxB.mic nThrds 336 size 1000 elapsedtime 0.0446899 GFlop/s 44.7528
```

On the MIC you can see the processes being executed as shown below.

(HPMC9) 03:21 root@mic0: ~ # pstree -up

```
(HPMC9) 03:21 root@mic0: ~ #  pstree -up
init(1)-+-coi_daemon(4354,micuser)-+-AxB.mic(25071)-+-{AxB.mic}(25077)
        |                          |                |-{AxB.mic}(25078)
        |                          |                |-{AxB.mic}(25079)
        |                          |                |-{AxB.mic}(25080)
        |                          |                |-{AxB.mic}(25081)
        |                          |                |-{AxB.mic}(25082)

 ...

        |                          |                |-{AxB.mic}(25335)
        |                          |                |-{AxB.mic}(25336)
        |                          |                |-{AxB.mic}(25337)
        |                          |                |-{AxB.mic}(25338)
        |                          |                |-{AxB.mic}(25339)
        |                          |                `-{AxB.mic}(25340)
        |                          |-{coi_daemon}(4356)
        |                          `-{coi_daemon}(25073)
        |-getty(4370)
        |-klogd(4333)
        |-mpssd(4339)-+-{mpssd}(4341)
        |             `-{mpssd}(4342)
        |-portmap(4290,1)
        |-sshd(4322)-+-sshd(4382)---bash(4384)---dint(23639)---cat(25342)
        |            |-sshd(13638)---bash(13640)---dcpus(23643)
        |            `-sshd(13642)---bash(13644)---pstree(25341)
        |-syslogd(4330)
        `-udevd(4113)-+-udevd(4336)
                      `-udevd(4363)
```

■ Interrupt and Stat Activities of a Program Offloaded to MIC

As the program is executed in a loop, and it is being offloaded to the MIC, we can reveal the stats of the cores. We use dcpus to display the stats by looping 24 times with a delay of 3 seconds in between. Since MIC has 228 cores, we use the -prune 6,3 option to display only the first six cores and the last three cores. Notice how the system and user usage went from 0 to higher numbers on the top most cores. As the loop is running, and more threads are allocated, the system usage is also increased on tne upper cores.

(HPMC9) 03:30 root@mic0: /tools # ./dcpus -transpose -iter 24 -delay 3 -prune 6,3 -stats user,system,idle,softirq

This command shows the specific stats (user,system,idle,softirq) on the first six cores ans last three cores (-prune 6,3) that are being transposed. The command is run in a loop iterating 24 times.

```
(HPMC9) 03:30 root@mic0: /tools #   ./dcpus -transpose -iter 24 -delay 3 -prune 6,3 -stats user,system,idle,
                                    user    system      idle   softirq
2026-04-26 03:30:57       CPU0         0         0       141         0
2026-04-26 03:30:57       CPU1         0         0       194         1
2026-04-26 03:30:57       CPU2         0         0       191         0
2026-04-26 03:30:57       CPU3         0         0       192         0
2026-04-26 03:30:57       CPU4         0         0       202         0
2026-04-26 03:30:57       CPU5         0         0       192         0
                            ...
2026-04-26 03:30:57     CPU225         0         0        98         0
2026-04-26 03:30:57     CPU226         0         0        99         0
2026-04-26 03:30:57     CPU227         0         0       113         0
                        summed         3        39     33396        17
used=59  idle=33396  usage 0.1763562995068%
                                    user    system      idle   softirq
2026-04-26 03:31:00       CPU0         0         0       344         0
2026-04-26 03:31:00       CPU1         0         0       299         1
2026-04-26 03:31:00       CPU2         0         0       306         0
2026-04-26 03:31:00       CPU3         0         0       303         0
2026-04-26 03:31:00       CPU4         0         0       299         0
2026-04-26 03:31:00       CPU5         0         0       306         0
                            ...
2026-04-26 03:31:00     CPU225         0         0       399         0
2026-04-26 03:31:00     CPU226         0         0       391         0
2026-04-26 03:31:00     CPU227         0         0       388         0
                        summed        13        84     78386        45
used=142  idle=78386  usage 0.180827220863896%
                                    user    system      idle   softirq
2026-04-26 03:31:03       CPU0         0         1       349         0
2026-04-26 03:31:03       CPU1         2       231       164         0
2026-04-26 03:31:03       CPU2         0         0       394         0
2026-04-26 03:31:03       CPU3         0         0       403         0
2026-04-26 03:31:03       CPU4         0         0       404         0
2026-04-26 03:31:03       CPU5         5       225       172         0
                            ...
2026-04-26 03:31:03     CPU225         0        80       238         0
2026-04-26 03:31:03     CPU226         0        41       284         0
2026-04-26 03:31:03     CPU227         1         2       305         0
                        summed       125      5630     74393        34
used=5789  idle=74393  usage 7.21982489835624%
                                    user    system      idle   softirq
2026-04-26 03:31:07       CPU0         0        35       307         4
2026-04-26 03:31:07       CPU1        73       104       128         0
2026-04-26 03:31:07       CPU2         0         0       298         1
2026-04-26 03:31:07       CPU3         0         0       297         0
2026-04-26 03:31:07       CPU4         0         0       297         0
2026-04-26 03:31:07       CPU5        55       126       119         0
                            ...
2026-04-26 03:31:07     CPU225         0        80       315         0
2026-04-26 03:31:07     CPU226         0        40       355         0
2026-04-26 03:31:07     CPU227         0         0       414         0
                        summed      1745      2709     72220        39
used=4493  idle=72220  usage 5.8568951807386%
                                    user    system      idle   softirq
2026-04-26 03:31:10       CPU0         0       125       222         0
2026-04-26 03:31:10       CPU1        10       293        85         0
2026-04-26 03:31:10       CPU2         0         1       309         1
2026-04-26 03:31:10       CPU3         0         0       307         0
2026-04-26 03:31:10       CPU4         0         0       307         0
2026-04-26 03:31:10       CPU5        11       284        99         0
                            ...
2026-04-26 03:31:10     CPU225         0         1       346         0
2026-04-26 03:31:10     CPU226         0         0       340         2
2026-04-26 03:31:10     CPU227         0         0       284         0
                        summed       683     13723     67670        19
used=14425  idle=67670  usage 17.5711066447409%
                                    user    system      idle   softirq
2026-04-26 03:31:14       CPU0         0        45       306         0
2026-04-26 03:31:14       CPU1        34       170       147         0
2026-04-26 03:31:14       CPU2         0       168       268         0
2026-04-26 03:31:14       CPU3         0         0       402         0
2026-04-26 03:31:14       CPU4         0         0       391         0
2026-04-26 03:31:14       CPU5        34       166       150         1
```

```
                           ...
2026-04-26 03:31:14     CPU225          0         2       349         0
2026-04-26 03:31:14     CPU226          0        81       253         1
2026-04-26 03:31:14     CPU227          0         0       401         0
                        summed       1575     12232     67384        28
used=13841  idle=67384  usage 17.0403200984918%
                                     user    system      idle   softirq
2026-04-26 03:31:17       CPU0          0       113       234         2
2026-04-26 03:31:17       CPU1         52       155        81         0
2026-04-26 03:31:17       CPU2         47       160        55         0
2026-04-26 03:31:17       CPU3          0         0       299         0
2026-04-26 03:31:17       CPU4          0         0       310         0
2026-04-26 03:31:17       CPU5         48       156        42         0
                           ...
2026-04-26 03:31:17     CPU225          9         0       340         0
2026-04-26 03:31:17     CPU226          1        88       279         2
2026-04-26 03:31:17     CPU227          0         0       362         0
                        summed       3167     11943     60548        13
used=15123  idle=60548  usage 19.9851990855149%
                                     user    system      idle   softirq
2026-04-26 03:31:21       CPU0          0       127       221         0
2026-04-26 03:31:21       CPU1          3       296       107         0
2026-04-26 03:31:21       CPU2          4       292       137         0
2026-04-26 03:31:21       CPU3          0         0       395         0
2026-04-26 03:31:21       CPU4          0         0       395         0
2026-04-26 03:31:21       CPU5          1       294       151         0
                           ...
2026-04-26 03:31:21     CPU225          0         2       343         0
2026-04-26 03:31:21     CPU226          8         0       338         0
2026-04-26 03:31:21     CPU227          0        48       289         0
                        summed        572     28079     55538         9
used=28660  idle=55538  usage 34.0388132734744%
                                     user    system      idle   softirq
2026-04-26 03:31:24       CPU0          0       126       229         0
2026-04-26 03:31:24       CPU1         51       132       173         0
2026-04-26 03:31:24       CPU2         53       123       179         0
2026-04-26 03:31:24       CPU3          2       122       273         0
2026-04-26 03:31:24       CPU4          0         0       396         0
2026-04-26 03:31:24       CPU5         38       137       179         0
                           ...
2026-04-26 03:31:24     CPU225          0         0       358         0
2026-04-26 03:31:24     CPU226          0         2       355         0
2026-04-26 03:31:24     CPU227          0         0       335         0
                        summed       4381     15568     62536        11
used=20015  idle=62536  usage 24.2456178604741%
                                     user    system      idle   softirq
2026-04-26 03:31:28       CPU0          0       156       192         1
2026-04-26 03:31:28       CPU1         37       167       143         0
2026-04-26 03:31:28       CPU2         56       133       160         0
2026-04-26 03:31:28       CPU3         49       139       160         0
2026-04-26 03:31:28       CPU4          0         0       329         0
2026-04-26 03:31:28       CPU5         47       141       160         0
                           ...
2026-04-26 03:31:28     CPU225         15         1       334         0
2026-04-26 03:31:28     CPU226          0        89       261         0
2026-04-26 03:31:28     CPU227          0         4       377         0
                        summed       4855     18524     58617         9
used=23464  idle=58617  usage 28.5863963645667%
                                     user    system      idle   softirq
2026-04-26 03:31:31       CPU0         16        49       284         0
2026-04-26 03:31:31       CPU1         69       277         2         0
2026-04-26 03:31:31       CPU2         61       260        28         0
2026-04-26 03:31:31       CPU3         94       253         2         0
2026-04-26 03:31:31       CPU4          0         0       367         1
2026-04-26 03:31:31       CPU5         45       276        27         0
                           ...
2026-04-26 03:31:31     CPU225          0        40       309         0
2026-04-26 03:31:31     CPU226          1        41       307         0
2026-04-26 03:31:31     CPU227          0        87       259         0
                        summed       4274     44741     28527        20
used=49094  idle=28527  usage 63.2483477409464%
                                     user    system      idle   softirq
2026-04-26 03:31:35       CPU0          6        42       305         0
2026-04-26 03:31:35       CPU1         24       135       194         0
2026-04-26 03:31:35       CPU2         24       126       203         0
2026-04-26 03:31:35       CPU3         23       127       203         0
2026-04-26 03:31:35       CPU4          0         0       305         0
2026-04-26 03:31:35       CPU5         24       128       201         0
                           ...
2026-04-26 03:31:35     CPU225          1        81       270         0
2026-04-26 03:31:35     CPU226          0        43       295         0
2026-04-26 03:31:35     CPU227          2         0       344         9
                        summed       3343     21450     56278        18
used=24811  idle=56278  usage 30.5972450024048%
                                     user    system      idle   softirq
2026-04-26 03:31:38       CPU0         11        42       298         0
2026-04-26 03:31:38       CPU1         69       159        65         0
2026-04-26 03:31:38       CPU2         64       162       125         0
2026-04-26 03:31:38       CPU3         59       168       121         0
2026-04-26 03:31:38       CPU4          0         0       401         0
2026-04-26 03:31:38       CPU5         35       190       123         0
                           ...
```

```
2026-04-26 03:31:38     CPU225          0        42       310         0
2026-04-26 03:31:38     CPU226          7        46       313         0
2026-04-26 03:31:38     CPU227          1       117       234         0
                        summed       6534     31213     35642         8
used=37930  idle=35642  usage 51.5549393791116%
                                      user    system      idle   softirq
2026-04-26 03:31:42       CPU0          0        81       267         1
2026-04-26 03:31:42       CPU1          6       293       105         0
2026-04-26 03:31:42       CPU2         62       191        85         0
2026-04-26 03:31:42       CPU3         31       259        60         0
2026-04-26 03:31:42       CPU4         44       235        59         0
2026-04-26 03:31:42       CPU5          1       289        59         0
                            ...
2026-04-26 03:31:42     CPU225         10         2       316         0
2026-04-26 03:31:42     CPU226          0        48       299         0
2026-04-26 03:31:42     CPU227          0        45       283         0
                        summed       1263     54167     31293        12
used=55442  idle=31293  usage 63.921139101862%
                                      user    system      idle   softirq
2026-04-26 03:31:45       CPU0          2       123       227         0
2026-04-26 03:31:45       CPU1         68       116       167         0
2026-04-26 03:31:45       CPU2         24       103       235         0
2026-04-26 03:31:45       CPU3         66       105       183         0
2026-04-26 03:31:45       CPU4         22       104       236         0
2026-04-26 03:31:45       CPU5         42       130       179         0
                            ...
2026-04-26 03:31:45     CPU225          0         1       369         2
2026-04-26 03:31:45     CPU226          0         2       351         0
2026-04-26 03:31:45     CPU227          0         0       305         0
                        summed       7521     26572     47220        12
used=34184  idle=47220  usage 41.993022455899%
                                      user    system      idle   softirq
2026-04-26 03:31:49       CPU0         14       122       214         0
2026-04-26 03:31:49       CPU1         20       211        74         0
2026-04-26 03:31:49       CPU2         24       208       118         0
2026-04-26 03:31:49       CPU3         33       198        74         0
2026-04-26 03:31:49       CPU4         56       176        62         0
2026-04-26 03:31:49       CPU5         21       211       118         0
                            ...
2026-04-26 03:31:49     CPU225          1         0       345         0
2026-04-26 03:31:49     CPU226          1        88       240         0
2026-04-26 03:31:49     CPU227          0         1       399         0
                        summed       8312     39342     21404         2
used=47776  idle=21404  usage 69.0604220873085%
                                      user    system      idle   softirq
2026-04-26 03:31:52       CPU0          9        85       254         0
2026-04-26 03:31:52       CPU1         24       268       100         0
2026-04-26 03:31:52       CPU2          8       168       172         0
2026-04-26 03:31:52       CPU3          9       208       175         0
2026-04-26 03:31:52       CPU4          8       218       176         0
2026-04-26 03:31:52       CPU5         19       242        87         0
                            ...
2026-04-26 03:31:52     CPU225          0         2       350         0
2026-04-26 03:31:52     CPU226          0       130       220         0
2026-04-26 03:31:52     CPU227          0         0       354        12
                        summed       2215     45485     42446        19
used=47735  idle=42446  usage 52.9324358789545%
                                      user    system      idle   softirq
2026-04-26 03:31:56       CPU0          0       134       193         8
2026-04-26 03:31:56       CPU1          4       215       128         0
2026-04-26 03:31:56       CPU2          0       202       143         0
2026-04-26 03:31:56       CPU3          0       188       154         0
2026-04-26 03:31:56       CPU4          0       165       150         0
2026-04-26 03:31:56       CPU5          1       202       146         0
                            ...
2026-04-26 03:31:56     CPU225          9        40       300         0
2026-04-26 03:31:56     CPU226         14        21       331         0
2026-04-26 03:31:56     CPU227          0        49       228         0
                        summed        136     41654     35033        46
used=41836  idle=35033  usage 54.4250608177549%
                                      user    system      idle   softirq
2026-04-26 03:31:59       CPU0          1        42       303        16
2026-04-26 03:31:59       CPU1         27       160       166         0
2026-04-26 03:31:59       CPU2          8       107       239         0
2026-04-26 03:31:59       CPU3         10       105       242         0
2026-04-26 03:31:59       CPU4          9       104       270         0
2026-04-26 03:31:59       CPU5         26       116       208         0
                            ...
2026-04-26 03:31:59     CPU225          8        46       293         0
2026-04-26 03:31:59     CPU226          4         1       346         1
2026-04-26 03:31:59     CPU227          0        40       382         0
                        summed       2824     24286     55308        32
used=27150  idle=55308  usage 32.9258531616095%
                                      user    system      idle   softirq
2026-04-26 03:32:03       CPU0          8        45       294         5
2026-04-26 03:32:03       CPU1         36       144       174         0
2026-04-26 03:32:03       CPU2         32       110       209         0
2026-04-26 03:32:03       CPU3         11        84       258         0
2026-04-26 03:32:03       CPU4         25        57       271         0
2026-04-26 03:32:03       CPU5         43       110       200         0
                            ...
2026-04-26 03:32:03     CPU225          0       121       227         4
```

```
2026-04-26 03:32:03     CPU226          0         2       348         3
2026-04-26 03:32:03     CPU227          1         0       330         0
                        summed      4861     16626     58879        50
used=21580  idle=58879  usage 26.8211138592327%
                                    user    system      idle   softirq
2026-04-26 03:32:06       CPU0        10        42       253        36
2026-04-26 03:32:06       CPU1        37       221        65         0
2026-04-26 03:32:06       CPU2        33       204       112         0
2026-04-26 03:32:06       CPU3        23       140       178         1
2026-04-26 03:32:06       CPU4        26       126       197         0
2026-04-26 03:32:06       CPU5        35       194       121         0
                            ...
2026-04-26 03:32:06     CPU225        10        41       258         4
2026-04-26 03:32:06     CPU226         0       134       214         0
2026-04-26 03:32:06     CPU227         0         0       317         6
                        summed      6416     37244     28776        63
used=43724  idle=28776  usage 60.3089655172414%
                                    user    system      idle   softirq
2026-04-26 03:32:10       CPU0         9       130       199         9
2026-04-26 03:32:10       CPU1        46       211       119         0
2026-04-26 03:32:10       CPU2        39       207       104         0
2026-04-26 03:32:10       CPU3        20       141       192         0
2026-04-26 03:32:10       CPU4        22       179       146         0
2026-04-26 03:32:10       CPU5        45       206        98         0
                            ...
2026-04-26 03:32:10     CPU225         0         1       362         6
2026-04-26 03:32:10     CPU226         0         0       314        16
2026-04-26 03:32:10     CPU227         0        50       328         4
                        summed      6301     39081     40555        58
used=45514  idle=40555  usage 52.8808281727451%
                                    user    system      idle   softirq
2026-04-26 03:32:13       CPU0         0        79       238         0
2026-04-26 03:32:13       CPU1         2        12       249         1
2026-04-26 03:32:13       CPU2         0         1       321         0
2026-04-26 03:32:13       CPU3         0         0       265         0
2026-04-26 03:32:13       CPU4         0         0       350         0
2026-04-26 03:32:13       CPU5         8         1       334         0
                            ...
2026-04-26 03:32:13     CPU225         0         0       322         0
2026-04-26 03:32:13     CPU226         0         0       350         0
2026-04-26 03:32:13     CPU227         0         0       295         0
                        summed        16       186     69814        22
used=224  idle=69814  usage 0.319826379965162%
                                    user    system      idle   softirq
2026-04-26 03:32:17       CPU0         0         0       354         0
2026-04-26 03:32:17       CPU1         0         0       398         1
2026-04-26 03:32:17       CPU2         0         0       332         0
2026-04-26 03:32:17       CPU3         0         0       390         0
2026-04-26 03:32:17       CPU4         0         0       298         0
2026-04-26 03:32:17       CPU5         0         0       301         3
                            ...
2026-04-26 03:32:17     CPU225         0         0       367         0
2026-04-26 03:32:17     CPU226         0         0       345         0
2026-04-26 03:32:17     CPU227         0         0       401         0
                        summed        16        86     79106        38
used=140  idle=79106  usage 0.17666506826843%
```

On the MIC we can reveal the interrupt activities as the the offloaded program is being run. As the program is executed in a loop, and it is being offloaded to the MIC, we can reveal the interrupt activities of the MIC. We use dint to display the interrupts (LOC,TLB,RES,TRM,CAL) by looping 24 times with a delay of 3 seconds in betwee. Since MIC has 228 cores, we use the -prune 6,3 option to display only the first six cores and the last three cores. Notice how the system interrupts of RES,LOC are being increased on the top most cores. As the loop is running, and more threads are allocated, the system interrupts also increased on tne upper cores.

(HPMC9) 03:30 root@mic0: /tools # ./dint -transpose -iter 24 -delay 3 -prune 6,3 -transpose -interrupts LOC,TLB,RES,TRM,CAL -sortby RES

```
#  ./dint -transpose -iter 24 -delay 3 -prune 6,3 -transpose -interrupts LOC,TLB,RES,TRM,CAL -sortby RES
                                  LOC       TLB       RES       TRM       CAL
2026-04-26 03:30:54   CPU30        53         0        10         0         0
2026-04-26 03:30:54   CPU29        53         0         6         0         0
2026-04-26 03:30:54    CPU0        33         0         2         0         0
2026-04-26 03:30:54  CPU222        12         0         1         0         0
2026-04-26 03:30:54  CPU225        13         0         1         0         0
2026-04-26 03:30:54   CPU26        54         1         1         0         0
                        ...
2026-04-26 03:30:54   CPU90        12         0         0         0         0
2026-04-26 03:30:54   CPU32        12         0         0         0         0
2026-04-26 03:30:54  CPU127        12         0         0         0         0
                     summed      2994         2        24         0        21

                                  LOC       TLB       RES       TRM       CAL
2026-04-26 03:30:57   CPU38        70         0        11         0         0
2026-04-26 03:30:57   CPU54       102         1        10         0         0
```

```
2026-04-26 03:30:57   CPU62        91         0         9         0         0
2026-04-26 03:30:57  CPU225        30         0         8         0         0
2026-04-26 03:30:57  CPU226        27         0         8         0         0
2026-04-26 03:30:57    CPU0        79         0         7         0         0
                         ...
2026-04-26 03:30:57   CPU90        29         0         0         0         0
2026-04-26 03:30:57   CPU32        30         0         0         0         0
2026-04-26 03:30:57  CPU127        29         0         0         0         0
                     summed      7765         9        80         0        23

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:01  CPU226        79         2        25         0         3
2026-04-26 03:31:01    CPU0       121         2        24         0         0
2026-04-26 03:31:01  CPU225        64        57        22         0         3
2026-04-26 03:31:01  CPU227        80         1        22         0         3
2026-04-26 03:31:01    CPU6        78         1        10         0         3
2026-04-26 03:31:01   CPU78        71         0         9         0         3
                         ...
2026-04-26 03:31:01  CPU190        29         0         0         0         3
2026-04-26 03:31:01  CPU200        29         0         0         0         3
2026-04-26 03:31:01  CPU206        29         0         0         0         3
                     summed      7510        72       314         0       715

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:04   CPU21       344         2      4449         0         0
2026-04-26 03:31:04   CPU29       346         2      4396         0         0
2026-04-26 03:31:04   CPU13       344         2      4391         0         0
2026-04-26 03:31:04   CPU89       339         2      4227         0         0
2026-04-26 03:31:04   CPU45       348         1      4189         0         0
2026-04-26 03:31:04   CPU41       344         2      4167         0         0
                         ...
2026-04-26 03:31:04   CPU90        29         1         0         0         0
2026-04-26 03:31:04   CPU32        29         1         0         0         0
2026-04-26 03:31:04  CPU127        28         1         0         0         0
                     summed     14682       317     95269         0       172

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:08   CPU21       173         9      2166         0         8
2026-04-26 03:31:08   CPU29       175         9      2122         0         8
2026-04-26 03:31:08   CPU17       174         9      2115         0         8
2026-04-26 03:31:08   CPU13       171         9      1791         0         8
2026-04-26 03:31:08    CPU5       174         8      1585         0         8
2026-04-26 03:31:08   CPU25       174         8      1485         0         8
                         ...
2026-04-26 03:31:08   CPU90        29         1         1         0         8
2026-04-26 03:31:08   CPU32        28         1         1         0         8
2026-04-26 03:31:08  CPU127        28         1         1         0         8
                     summed     12909       568     31386         0      1914

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:11   CPU33       284         8      1661         0         2
2026-04-26 03:31:11   CPU97       283         6      1625         0         2
2026-04-26 03:31:11   CPU53       283         8      1616         0         2
2026-04-26 03:31:11   CPU73       279         7      1607         0         2
2026-04-26 03:31:11  CPU113       284         7      1562         0         2
2026-04-26 03:31:11  CPU105       285         8      1558         0         2
                         ...
2026-04-26 03:31:11  CPU118        29         0         0         0         2
2026-04-26 03:31:11   CPU90        29         0         0         0         2
2026-04-26 03:31:11  CPU127        29         0         0         0         2
                     summed     19617       372     60225         0       590

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:15  CPU161       302         2      1324         0         6
2026-04-26 03:31:15    CPU1       333        72      1322         0         5
2026-04-26 03:31:15  CPU165       312         2      1312         0         5
2026-04-26 03:31:15  CPU197       309         2      1310         0         5
2026-04-26 03:31:15  CPU157       309         2      1305         0         5
2026-04-26 03:31:15  CPU173       307         2      1300         0         5
                         ...
2026-04-26 03:31:15   CPU90        29         1         1         0         5
2026-04-26 03:31:15   CPU32        30         1         1         0         5
2026-04-26 03:31:15  CPU127        29         1         1         0         5
                     summed     27565       559     83410         0      1429

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:18    CPU6       228       107       660         0         8
2026-04-26 03:31:18  CPU173       191        10       646         0         8
2026-04-26 03:31:18  CPU161       192        10       642         0         8
2026-04-26 03:31:18  CPU105       193         9       638         0         8
2026-04-26 03:31:18    CPU1       279        99       628         0         8
2026-04-26 03:31:18  CPU181       194        11       621         0         8
                         ...
2026-04-26 03:31:18  CPU206        29         1         1         0         8
2026-04-26 03:31:18   CPU32        29         1         1         0         8
2026-04-26 03:31:18  CPU127        29         1         1         0         8
                     summed     21718      1226     28897         0      1988

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:22   CPU45       272        13       937         0         2
2026-04-26 03:31:22   CPU41       260        12       934         0         2
2026-04-26 03:31:22   CPU61       271        14       923         0         2
```

```
2026-04-26 03:31:22   CPU53       268        11       880         0         2
2026-04-26 03:31:22   CPU33       271        12       871         0         2
2026-04-26 03:31:22   CPU25       272        12       832         0         2
                        ...
2026-04-26 03:31:22  CPU206        29         0         0         0         2
2026-04-26 03:31:22   CPU32        30         0         0         0         2
2026-04-26 03:31:22  CPU127        29         0         0         0         2
                     summed     30634      1244     62036         0       708

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:26    CPU1       347       121      1013         0         6
2026-04-26 03:31:26    CPU6       337       124       821         0         6
2026-04-26 03:31:26   CPU10       298        92       820         0         6
2026-04-26 03:31:26  CPU105       285         2       762         0         6
2026-04-26 03:31:26   CPU69       284         2       726         0         6
2026-04-26 03:31:26  CPU102       287         2       717         0         6
                        ...
2026-04-26 03:31:26   CPU71        28         1         1         0         6
2026-04-26 03:31:26   CPU32        28         1         1         0         6
2026-04-26 03:31:26  CPU127        28         1         1         0         6
                     summed     37988       693     52706         0      1645

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:29    CPU1       274       146       747         0         7
2026-04-26 03:31:29  CPU109       237        10       664         0         7
2026-04-26 03:31:29   CPU10       250       128       605         0         7
2026-04-26 03:31:29    CPU6       249       160       582         0         7
2026-04-26 03:31:29  CPU105       237        10       553         0         7
2026-04-26 03:31:29  CPU125       239        12       446         0         7
                        ...
2026-04-26 03:31:29   CPU16        29         1         1         0         7
2026-04-26 03:31:29  CPU200        29         1         1         0         7
2026-04-26 03:31:29   CPU32        29         1         1         0         7
                     summed     36162      2072     31776         0      1739

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:33  CPU102       264         8       612         0         7
2026-04-26 03:31:33  CPU165       265         7       558         0         7
2026-04-26 03:31:33  CPU185       265         8       546         0         7
2026-04-26 03:31:33  CPU161       264         8       536         0         7
2026-04-26 03:31:33  CPU198       265         7       518         0         7
2026-04-26 03:31:33   CPU90       292         8       510         0         7
                        ...
2026-04-26 03:31:33  CPU163        32         0         0         0         7
2026-04-26 03:31:33  CPU220        32         0         0         0         7
2026-04-26 03:31:33  CPU200        32         0         0         0         7
                     summed     41379      1387     49359         0      1838

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:37    CPU1       348       167       907         0         0
2026-04-26 03:31:37   CPU10       349       166       767         0         0
2026-04-26 03:31:37  CPU137       338         2       674         0         0
2026-04-26 03:31:37  CPU145       338         2       671         0         0
2026-04-26 03:31:37  CPU214       338         3       656         0         0
2026-04-26 03:31:37  CPU222       339         2       621         0         0
                        ...
2026-04-26 03:31:37   CPU44        29         1         0         0         0
2026-04-26 03:31:37   CPU16        29         1         0         0         0
2026-04-26 03:31:37   CPU32        30         1         0         0         0
                     summed     58745       884     71948         0       324

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:40    CPU1       292       194       708         0         8
2026-04-26 03:31:40   CPU10       278       199       656         0         8
2026-04-26 03:31:40    CPU6       290       139       382         0         8
2026-04-26 03:31:40   CPU68       178         2       278         0         8
2026-04-26 03:31:40   CPU62       234        12       278         0         8
2026-04-26 03:31:40   CPU63       234        12       277         0         8
                        ...
2026-04-26 03:31:40  CPU148        28         1         1         0         8
2026-04-26 03:31:40  CPU220        28         1         1         0         8
2026-04-26 03:31:40  CPU200        28         1         1         0         8
                     summed     45793      2878     25259         0      1988

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:44   CPU53       250        11       558         0         8
2026-04-26 03:31:44   CPU85       250        11       531         0         8
2026-04-26 03:31:44  CPU141       251        11       513         0         8
2026-04-26 03:31:44  CPU153       248         6       501         0         8
2026-04-26 03:31:44  CPU137       250        11       469         0         8
2026-04-26 03:31:44  CPU129       251         7       452         0         8
                        ...
2026-04-26 03:31:44  CPU148        29         0         0         0         8
2026-04-26 03:31:44  CPU220        29         0         0         0         8
2026-04-26 03:31:44  CPU200        29         0         0         0         8
                     summed     48171      2297     49533         0      2032

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:47    CPU1       356       224       971         0         2
2026-04-26 03:31:47   CPU10       355       222       959         0         2
2026-04-26 03:31:47    CPU6       355       131       631         0         2
2026-04-26 03:31:47   CPU14       354        96       550         0         2
```

```
2026-04-26 03:31:47  CPU118       337        15       547         0         3
2026-04-26 03:31:47    CPU2       357        15       485         0         2
                         ...
2026-04-26 03:31:47  CPU204        29         1         1         0         2
2026-04-26 03:31:47  CPU220        29         1         1         0         2
2026-04-26 03:31:47  CPU200        29         1         1         0       204
                     summed     71825      3748     58916         0       786

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:51    CPU1       310       227       575         0         5
2026-04-26 03:31:51   CPU10       197       237       560         0         5
2026-04-26 03:31:51    CPU6       262       180       490         0         5
2026-04-26 03:31:51   CPU14       258        55       270         0         5
2026-04-26 03:31:51   CPU34       208         2       203         0         5
2026-04-26 03:31:51   CPU44       211         3       200         0         5
                         ...
2026-04-26 03:31:51    CPU0       209         2        40         0         5
2026-04-26 03:31:51  CPU226       160        60        26         0         4
2026-04-26 03:31:51  CPU227       134         3        24         0       184
                     summed     47564      1256     25065         0      1363

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:55    CPU1       314       243       932         0         8
2026-04-26 03:31:55    CPU6       258       273       733         0         8
2026-04-26 03:31:55   CPU10       285       242       565         0         8
2026-04-26 03:31:55   CPU21       272         7       451         0         8
2026-04-26 03:31:55   CPU53       271         7       423         0         8
2026-04-26 03:31:55   CPU49       272         7       416         0         8
                         ...
2026-04-26 03:31:55  CPU226       168         2        53         0        19
2026-04-26 03:31:55  CPU227       179         2        47         0        54
2026-04-26 03:31:55    CPU0       235        58        31         0        45
                     summed     50492      2039     37130         0      1933

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:31:58    CPU1       267       306       930         0         7
2026-04-26 03:31:58    CPU6       211       304       660         0         7
2026-04-26 03:31:58   CPU10       217       277       656         0         7
2026-04-26 03:31:58   CPU97       221         7       376         0         7
2026-04-26 03:31:58   CPU77       218         7       317         0         7
2026-04-26 03:31:58   CPU69       218         7       307         0         7
                         ...
2026-04-26 03:31:58    CPU0       223         5        26         0        15
2026-04-26 03:31:58  CPU226       130         2        26         0        30
2026-04-26 03:31:58  CPU227       130         0        25         0         5
                     summed     39673      2126     32380         0      1712

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:32:02    CPU1       299        20       475         0         4
2026-04-26 03:32:02  CPU209       281         4       338         0         4
2026-04-26 03:32:02   CPU93       274         4       328         0         4
2026-04-26 03:32:02  CPU181       278         4       320         0         4
2026-04-26 03:32:02   CPU69       274         4       316         0         4
2026-04-26 03:32:02   CPU49       276         4       296         0         4
                         ...
2026-04-26 03:32:02    CPU0       215         1        18         0        40
2026-04-26 03:32:02  CPU226        87         1        17         0        54
2026-04-26 03:32:02  CPU227       188         3        14         0        10
                     summed     46214       763     37389         0      1107

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:32:05    CPU1       318       278       939         0         5
2026-04-26 03:32:05    CPU6       338       331       787         0         5
2026-04-26 03:32:05   CPU10       311       293       671         0         5
2026-04-26 03:32:05   CPU33       279         7       362         0         5
2026-04-26 03:32:05   CPU45       280         7       280         0         5
2026-04-26 03:32:05   CPU49       282         7       271         0         5
                         ...
2026-04-26 03:32:05  CPU225       204         6        36         0         4
2026-04-26 03:32:05    CPU0       265         0        24         0         5
2026-04-26 03:32:05  CPU227        64         2        18         0       166
                     summed     53800      2270     40860         0      1335

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:32:09    CPU6       307       355       775         0         6
2026-04-26 03:32:09    CPU1       265       282       674         0         7
2026-04-26 03:32:09   CPU10       243       309       627         0         6
2026-04-26 03:32:09   CPU21       238         3       233         0         6
2026-04-26 03:32:09   CPU11       199         6       202         0         6
2026-04-26 03:32:09   CPU61       237         4       194         0         6
                         ...
2026-04-26 03:32:09  CPU226       172         2        31         0        27
2026-04-26 03:32:09    CPU0       198         7        28         0        16
2026-04-26 03:32:09  CPU227        97         3        25         0        79
                     summed     46771      1607     28257         0      1508

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:32:13    CPU1       152        23       246         0         4
2026-04-26 03:32:13  CPU129       121         4        90         0         4
2026-04-26 03:32:13   CPU98        91         4        88         0         4
2026-04-26 03:32:13  CPU162        91         4        83         0         4
2026-04-26 03:32:13  CPU193       152         4        78         0         4
```

```
2026-04-26 03:32:13   CPU34        93         4        75         0         4
                        ...
2026-04-26 03:32:13  CPU203        63         3        21         0         4
2026-04-26 03:32:13    CPU0       130         1         6         0         0
2026-04-26 03:32:13  CPU227        69         0         4         0         4
                     summed     19147       803      8377         0       932

                                   LOC       TLB       RES       TRM       CAL
2026-04-26 03:32:16   CPU14        71         0        10         0         0
2026-04-26 03:32:16   CPU22        74         0         9         0         0
2026-04-26 03:32:16   CPU13        71         0         6         0         0
2026-04-26 03:32:16  CPU225        42         0         6         0         0
2026-04-26 03:32:16  CPU226        58         0         6         0         0
2026-04-26 03:32:16    CPU0        31         0         4         0         0
                        ...
2026-04-26 03:32:16   CPU90        29         0         0         0         0
2026-04-26 03:32:16   CPU32        29         0         0         0         0
2026-04-26 03:32:16  CPU127        29         0         0         0         0
                     summed      7068         4        53         0         8
```

2-5 Finding Primes Offloading to MIC

This chapter shows how to use your HPMC calculator to find the prime numbers using openMP. The operations will be offloaded to the Intel Xeon Phi MIC.

Consider the program prime3a.c shown in the following listing:

-- Program Code 2.5.1 : [LISTING prime3a.c] - [Finding Primes using openMP]

(raw text)

```
1.     #include <stdio.h>
2.     #include <stdlib.h>
3.     #include <string.h>
4.     #include <math.h>
5.     #include <omp.h>
6.
7.     // fill p[] with primes up to max plus one more
8.     int
9.     primearray(int max, int p[]) {
10.        p[0] = 3;
11.        int top = 0, n, i;
12.        for (n = 5;; n += 2)
13.            for (i = 0;; i++)
14.                if (n % p[i] == 0)
15.                    break;
16.                else
17.                    if (p[i]*p[i] > n) {
18.                        p[++top] = n;
19.        printf("prime: %d top: %d %d %ld  \n",
20.                 n, top, p[top], (long)p[top]*p[top]);
21.                        if (n > max)
22.                            return top; // last n
23.                        break;
24.                    }
25.    }
26.    // use primes in p[] to check up to max
27.    // 5 is p[1] is the 3rd prime, so start count from 2
28.    int
29.    primecount_omp(long max, int p[]) {
30.        int n, i, sum = 2;
31.        #pragma omp parallel for private(i) schedule(guided)
       reduction(+:sum)
32.        for (n = 5; n <= max; n += 2)
33.            for (i = 0;; i++)
34.                if (n % p[i] == 0)
35.                    break;
36.                else
37.                    if (p[i]*p[i] > n) {
38.                        sum++;
39.                        break;
40.                    }
41.        return sum;
42.    }
43.    // call primearray() with SQR and then primecount_omp() with MAX
44.    // PNT (x/logx) to find approx. size of array p[] psize
45.    int
46.    main(int argc, char** argv) {
47.        if (argc < 2) {
48.            printf("You need to supply a limit\n");
49.            return 1;
50.        }
51.        long MAX = strtoul(argv[1], 0, 10);
52.        int SQR = sqrt(MAX);
53.        int psize = 50 + 1.2 * SQR / log(SQR);
54.        int *p = malloc(psize * sizeof*p);
55.        int top = primearray(SQR, p);
56.        printf("psize: %d top: %d %d %ld\n", psize, top, p[top],
       (long)p[top]*p[top]);
```

```
57.         int sum = primecount_omp(MAX, p);
58.         printf("Count: %d (%.2f%%)\n", sum, (double) 100*sum / MAX) ;
59.         double lnx = MAX / log(MAX);
60.         printf("LnX:   %.2f (%f)\n",  lnx, sum / lnx);
61.         return 0;
62.     }
```


Compile prime3a.c to be executed on the MIC:

icc -O3 -mmic -qopenmp -L /opt/intel/lib/mic -Wno-unknown-pragmas -std=c99 -liomp5 prime3a.c -o prime3a.mic

Running the prime3a.phi by offloading it to the MIC. The upper limit selected is 100000000.

02:28 root@HPMC9: /mm03fs/PHI1/BSR # xphi prime3a.phi -a 100000000

```
02:28 root@HPMC9: /mm03fs/PHI1/BSR #  xphi prime3a.phi -a 100000000
prime: 5 top: 1 5 25
prime: 7 top: 2 7 49
prime: 11 top: 3 11 121
prime: 13 top: 4 13 169
prime: 17 top: 5 17 289
prime: 19 top: 6 19 361
prime: 23 top: 7 23 529
prime: 29 top: 8 29 841
prime: 31 top: 9 31 961
prime: 37 top: 10 37 1369
prime: 41 top: 11 41 1681
prime: 43 top: 12 43 1849
prime: 47 top: 13 47 2209
...
prime: 9929 top: 1222 9929 98585041
prime: 9931 top: 1223 9931 98624761
prime: 9941 top: 1224 9941 98823481
prime: 9949 top: 1225 9949 98982601
prime: 9967 top: 1226 9967 99341089
prime: 9973 top: 1227 9973 99460729
prime: 10007 top: 1228 10007 100140049
psize: 1352 top: 1228 10007 100140049
Count: 5761455 (5.76%)
LnX:   5428681.02 (1.061299)
```

On the MIC you can see the processes being executed as shown below.

(HPMC9) 02:28 root@mic0: ~ # pstree -up

```
(HPMC9) 02:28 root@mic0: ~ #  pstree -up
init(1)-+-coi_daemon(4353,micuser)-+-prime3a.phi(5543)-+-{prime3a.phi}(5549)
        |                          |                   |-{prime3a.phi}(5550)
        |                          |                   |-{prime3a.phi}(5551)
        |                          |                   |-{prime3a.phi}(5552)
        |                          |                   |-{prime3a.phi}(5553)
        |                          |                   |-{prime3a.phi}(5554)
        |                          |                   |-{prime3a.phi}(5555)
        |                          |                   |-{prime3a.phi}(5556)
        |                          |                   |-{prime3a.phi}(5557)
        |                          |                   |-{prime3a.phi}(5558)
        |                          |                   |-{prime3a.phi}(5559)
...
        |                          |                   |-{prime3a.phi}(5769)
        |                          |                   |-{prime3a.phi}(5770)
        |                          |                   |-{prime3a.phi}(5771)
        |                          |                   `-{prime3a.phi}(5772)
        |                          |-{coi_daemon}(4355)
        |                          `-{coi_daemon}(5545)
        |-getty(4369)
        |-klogd(4332)
        |-mpssd(4338)-+-{mpssd}(4340)
```

```
|             `-{mpssd}(4341)
|-portmap(4290,1)
|-sshd(4321)---sshd(5306)---bash(5308)---pstree(5773)
|-syslogd(4329)
`-udevd(4113)-+-udevd(4335)
              `-udevd(4362)
```

Chapter 3
CUDA Programming over the HPMC GPU

Image File

The HPMC is equipped with a GPU that allows you to do CUDA programming. CUDA (Compute Unified Device Architecture) is a parallel computing platform and programming model that makes using a GPU for general purpose computing simple.

Image File

Read Chapter 3
Back to Contents

3-1 CUDA Programming over the HPMC GPU

The HPMC computer has a GPU that allows you to run CUDA related programs. Your HPMC has been already configured with the essential device drivers so that the GPU can serve you to run any of the following: torch, pytorch, theano, caffe, cuda, and pycuda.

About your HPMC CUDA Device

checkDeviceInfor

```
20:00 root@HPMC7: /appz/professional-c-cuda-programming/CodeSamples/chapter02 #  ./checkDeviceInfor
./checkDeviceInfor Starting...
Detected 1 CUDA Capable device(s)
Device 0: "GeForce GTX TITAN X"
  CUDA Driver Version / Runtime Version          9.2 / 6.5
  CUDA Capability Major/Minor version number:    5.2
  Total amount of global memory:                 11.93 MBytes (12805668864 bytes)
  GPU Clock rate:                                1240 MHz (1.24 GHz)
  Memory Clock rate:                             3505 Mhz
  Memory Bus Width:                              384-bit
  L2 Cache Size:                                 3145728 bytes
  Max Texture Dimension Size (x,y,z)             1D=(65536), 2D=(65536,65536), 3D=(4096,4096,4096)
  Max Layered Texture Size (dim) x layers        1D=(16384) x 2048, 2D=(16384,16384) x 2048
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  2048
  Maximum number of threads per block:           1024
  Maximum sizes of each dimension of a block:    1024 x 1024 x 64
  Maximum sizes of each dimension of a grid:     2147483647 x 65535 x 65535
  Maximum memory pitch:                          2147483647 bytes
```

./simpleDeviceQuery

```
19:59 root@HPMC7: /appz/professional-c-cuda-programming/CodeSamples/chapter03 #  ./simpleDeviceQuery
Device 0: GeForce GTX TITAN X
  Number of multiprocessors:                     24
  Total amount of constant memory:               64.00 KB
  Total amount of shared memory per block:       48.00 KB
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per block:           1024
  Maximum number of threads per multiprocessor:  2048
  Maximum number of warps per multiprocessor:    64
```

To make sure that your GPU is ready to serve CUDA programming capability, just invoke the randomFog program at the prompt. Figure shows the randomFog. In case the randomFog failed to execute then refer to Appendix Troubleshooting the CUDA GPU.

full view

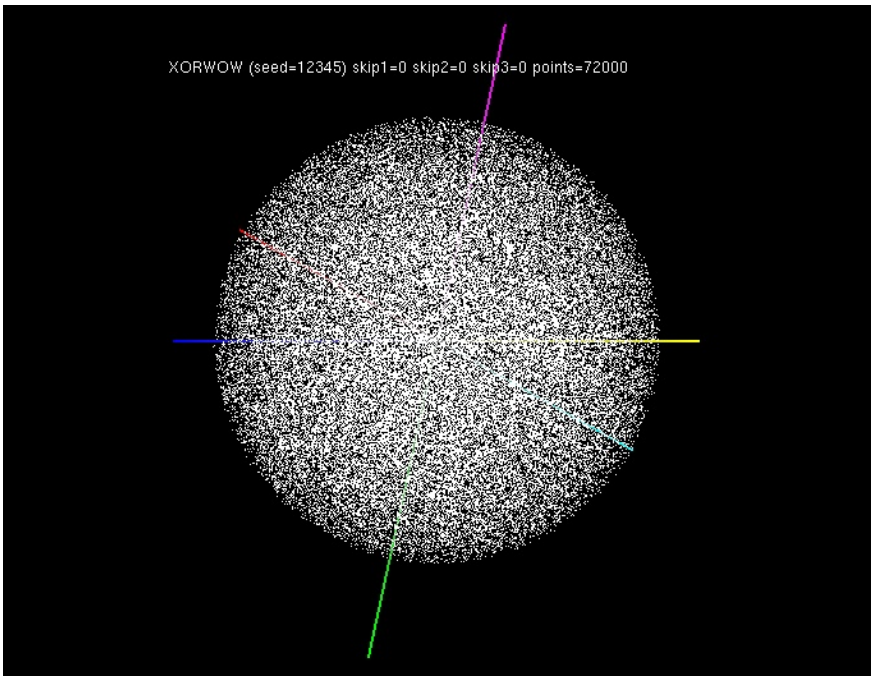


Figure. 3.1.1 [Random FOG running on the CUDA Capable GPU]

HPMC 2022 by Bassem Jamaleddine

■ CUDNN with Torch

You program in Lua and run torch programs on your HPMC.

torch: A Tensor library like NumPy, with strong GPU support PyTorch is used either as a replacement for NumPy to use the GPU as a processing agent. It is also used a platform for deep learning research.

20:09 root@HPMC7: /appz/torch # th bench.lua

```
20:09 root@HPMC7: /appz/torch #  th bench.lua
Found Environment variable CUDNN_PATH = /usr/local/cuda-9.2/lib64/libcudnn.so.7.1.4
Running on device: GeForce GTX TITAN X

CONFIG: input = 3x128x128 * ker = 3x96x11x11 (bs = 128, stride = 1)
cudnn.SpatialConvolution               :updateOutput():      51.10
cudnn.SpatialConvolution             :updateGradInput():      57.34
cudnn.SpatialConvolution           :accGradParameters():      50.86
cudnn.SpatialConvolution                        :TOTAL:     159.30
nn.SpatialConvolutionMM                :updateOutput():      44.83
nn.SpatialConvolutionMM              :updateGradInput():      54.59
nn.SpatialConvolutionMM            :accGradParameters():      56.69
nn.SpatialConvolutionMM                         :TOTAL:     156.12
ccn2.SpatialConvolution                :updateOutput():      32.38
ccn2.SpatialConvolution              :updateGradInput():      50.96
ccn2.SpatialConvolution            :accGradParameters():      44.72
ccn2.SpatialConvolution                         :TOTAL:     128.06

CONFIG: input = 64x64x64 * ker = 64x128x9x9 (bs = 128, stride = 1)
cudnn.SpatialConvolution               :updateOutput():      89.60
cudnn.SpatialConvolution             :updateGradInput():     264.64
cudnn.SpatialConvolution           :accGradParameters():     141.98
cudnn.SpatialConvolution                        :TOTAL:     496.21
nn.SpatialConvolutionMM                :updateOutput():     138.05
nn.SpatialConvolutionMM              :updateGradInput():     219.20
nn.SpatialConvolutionMM            :accGradParameters():     330.22
nn.SpatialConvolutionMM                         :TOTAL:     687.47
ccn2.SpatialConvolution                :updateOutput():     109.02
```

```
ccn2.SpatialConvolution              :updateGradInput():     125.64
ccn2.SpatialConvolution            :accGradParameters():     227.74
ccn2.SpatialConvolution                          :TOTAL:     462.40

CONFIG: input = 128x32x32 * ker = 128x128x9x9 (bs = 128, stride = 1)
cudnn.SpatialConvolution                 :updateOutput():      30.75
cudnn.SpatialConvolution               :updateGradInput():      74.42
cudnn.SpatialConvolution             :accGradParameters():      54.36
cudnn.SpatialConvolution                         :TOTAL:     159.52
nn.SpatialConvolutionMM                  :updateOutput():      53.73
nn.SpatialConvolutionMM                :updateGradInput():     103.11
nn.SpatialConvolutionMM              :accGradParameters():      55.03
nn.SpatialConvolutionMM                          :TOTAL:     211.87
ccn2.SpatialConvolution                  :updateOutput():      37.26
ccn2.SpatialConvolution                :updateGradInput():      46.71
ccn2.SpatialConvolution              :accGradParameters():      85.09
ccn2.SpatialConvolution                          :TOTAL:     169.07

CONFIG: input = 128x16x16 * ker = 128x128x7x7 (bs = 128, stride = 1)
cudnn.SpatialConvolution                 :updateOutput():       4.33
cudnn.SpatialConvolution               :updateGradInput():       7.33
cudnn.SpatialConvolution             :accGradParameters():       8.70
cudnn.SpatialConvolution                         :TOTAL:      20.36
nn.SpatialConvolutionMM                  :updateOutput():      14.75
nn.SpatialConvolutionMM                :updateGradInput():      18.78
nn.SpatialConvolutionMM              :accGradParameters():      12.61
nn.SpatialConvolutionMM                          :TOTAL:      46.14
ccn2.SpatialConvolution                  :updateOutput():       4.57
ccn2.SpatialConvolution                :updateGradInput():       4.29
ccn2.SpatialConvolution              :accGradParameters():       7.69
ccn2.SpatialConvolution                          :TOTAL:      16.55

CONFIG: input = 384x13x13 * ker = 384x384x3x3 (bs = 128, stride = 1)
cudnn.SpatialConvolution                 :updateOutput():       7.77
cudnn.SpatialConvolution               :updateGradInput():      11.71
cudnn.SpatialConvolution             :accGradParameters():      12.91
cudnn.SpatialConvolution                         :TOTAL:      32.39
nn.SpatialConvolutionMM                  :updateOutput():      13.15
nn.SpatialConvolutionMM                :updateGradInput():      17.28
nn.SpatialConvolutionMM              :accGradParameters():      14.63
nn.SpatialConvolutionMM                          :TOTAL:      45.06
ccn2.SpatialConvolution                  :updateOutput():       7.46
ccn2.SpatialConvolution                :updateGradInput():       8.58
ccn2.SpatialConvolution              :accGradParameters():      14.21
ccn2.SpatialConvolution                          :TOTAL:      30.25
```

■ nn with PyTorch

In PyTorch, the nn package defines a set of Modules, which are roughly equivalent to neural network layers. A Module receives input Tensors and computes output Tensors, but may also hold internal state such as Tensors containing learnable parameters. The nn package also defines a set of useful loss functions that are commonly used when training neural networks.

-- Program Code 3.1.1 : [LISTING pytnn.py] - [Implement polynomial model network using nn package]

(raw text)

```
1.     # -*- coding: utf-8 -*-
2.     import torch
3.     import math
4.
5.
6.     class LegendrePolynomial3(torch.autograd.Function):
7.         """
8.         We can implement our own custom autograd Functions by subclassing
9.         torch.autograd.Function and implementing the forward and backward
      passes
10.        which operate on Tensors.
11.        """
12.
13.        @staticmethod
14.        def forward(ctx, input):
15.            """
16.            In the forward pass we receive a Tensor containing the input
      and return
17.            a Tensor containing the output. ctx is a context object that
```

```
      can be used
18.           to stash information for backward computation. You can cache
      arbitrary
19.           objects for use in the backward pass using the
      ctx.save_for_backward method.
20.           """
21.           ctx.save_for_backward(input)
22.           return 0.5 * (5 * input ** 3 - 3 * input)
23.
24.       @staticmethod
25.       def backward(ctx, grad_output):
26.           """
27.           In the backward pass we receive a Tensor containing the
      gradient of the loss
28.           with respect to the output, and we need to compute the gradient
      of the loss
29.           with respect to the input.
30.           """
31.           input, = ctx.saved_tensors
32.           return grad_output * 1.5 * (5 * input ** 2 - 1)
33.
34.
35.    dtype = torch.float
36.    device = torch.device("cpu")
37.    # device = torch.device("cuda:0")  # Uncomment this to run on GPU
38.
39.    # Create Tensors to hold input and outputs.
40.    # By default, requires_grad=False, which indicates that we do not need
      to
41.    # compute gradients with respect to these Tensors during the backward
      pass.
42.    x = torch.linspace(-math.pi, math.pi, 2000, device=device, dtype=dtype)
43.    y = torch.sin(x)
44.
45.    # Create random Tensors for weights. For this example, we need
46.    # 4 weights: y = a + b * P3(c + d * x), these weights need to be
      initialized
47.    # not too far from the correct result to ensure convergence.
48.    # Setting requires_grad=True indicates that we want to compute
      gradients with
49.    # respect to these Tensors during the backward pass.
50.    a = torch.full((), 0.0, device=device, dtype=dtype, requires_grad=True)
51.    b = torch.full((), -1.0, device=device, dtype=dtype,
      requires_grad=True)
52.    c = torch.full((), 0.0, device=device, dtype=dtype, requires_grad=True)
53.    d = torch.full((), 0.3, device=device, dtype=dtype, requires_grad=True)
54.
55.    learning_rate = 5e-6
56.    for t in range(2000):
57.        # To apply our Function, we use Function.apply method. We alias
      this as 'P3'.
58.        P3 = LegendrePolynomial3.apply
59.
60.        # Forward pass: compute predicted y using operations; we compute
61.        # P3 using our custom autograd operation.
62.        y_pred = a + b * P3(c + d * x)
63.
64.        # Compute and print loss
65.        loss = (y_pred - y).pow(2).sum()
66.        if t % 100 == 99:
67.            print(t, loss.item())
68.
69.        # Use autograd to compute the backward pass.
70.        loss.backward()
71.
72.        # Update weights using gradient descent
73.        with torch.no_grad():
74.            a -= learning_rate * a.grad
75.            b -= learning_rate * b.grad
76.            c -= learning_rate * c.grad
77.            d -= learning_rate * d.grad
78.
79.            # Manually zero the gradients after updating weights
80.            a.grad = None
81.            b.grad = None
82.            c.grad = None
83.            d.grad = None
84.
85.    print(f'Result: y = {a.item()} + {b.item()} * P3({c.item()} +
      {d.item()} x)')
86.
87.
```

Running the program on the HPMC.

```
14:32 root@HPMC7: /home/hpcusr/examples/pytorch #  python3 pytnn.py
99 209.95834350585938
199 144.66018676757812
299 100.70249938964844
399 71.03519439697266
499 50.97850799560547
599 37.403133392333984
699 28.206867218017578
799 21.97318458557129
899 17.7457275390625
999 14.877889633178711
1099 12.93176555633545
1199 11.610918998718262
1299 10.71425724029541
1399 10.10548210144043
1499 9.692106246948242
1599 9.411375045776367
1699 9.220745086669922
1799 9.091285705566406
1899 9.003360748291016
1999 8.943639755249023
Result: y = -5.394172664097141e-09 + -2.208526849746704 * P3(1.367587154632588e-09 + 0.2554861009120941
```

3-2 CUDA Fortran

FORTRAN program can call CUDA routines by linking to CUDA object files. The following shows an example on how to compile a FORTRAN program that calls CUDA through an external kernel_wrapper method.

-- Program Code 3.2.1 : [LISTING fortest.f95] - [CUDA Program]

(raw text)

```
1.     #include <stdio.h>
2.     #include <stdlib.h>
3.     #include <string.h>
4.     #include <cuda.h>
5.     #include <cuda_runtime.h>
6.
7.
8.     // simple kernel function that adds two vectors
9.     __global__ void vect_add(float *a, float *b, int N)
10.    {
11.       int idx = threadIdx.x;
12.       if (idx<N) a[idx] = a[idx] + b[idx];
13.    }
14.
15.    // function called from main fortran program
16.    extern "C" void kernel_wrapper_(float *a, float *b, int *Np)
17.    {
18.       float  *a_d, *b_d;  // declare GPU vector copies
19.
20.       int blocks = 1;     // uses 1 block of
21.       int N = *Np;        // N threads on GPU
22.
23.       // Allocate memory on GPU
24.       cudaMalloc( (void **)&a_d, sizeof(float) * N );
25.       cudaMalloc( (void **)&b_d, sizeof(float) * N );
26.
27.       // copy vectors from CPU to GPU
28.       cudaMemcpy( a_d, a, sizeof(float) * N, cudaMemcpyHostToDevice );
29.       cudaMemcpy( b_d, b, sizeof(float) * N, cudaMemcpyHostToDevice );
30.
31.       // call function on GPU
32.       vect_add<<< blocks, N >>>( a_d, b_d, N);
33.
34.       // copy vectors back from GPU to CPU
35.       cudaMemcpy( a, a_d, sizeof(float) * N, cudaMemcpyDeviceToHost );
36.       cudaMemcpy( b, b_d, sizeof(float) * N, cudaMemcpyDeviceToHost );
37.
38.       // free GPU memory
39.       cudaFree(a_d);
40.       cudaFree(b_d);
41.       return;
42.    }
```

HPMC 2022

-- Program Code 3.2.2 : [LISTING fortest.f95] - [Fortran 95 Program]

(raw text)

```
1.     PROGRAM fortest
2.
3.     ! simple program which creates 2 vectors and adds them in a
4.     ! cuda function
5.
6.     IMPLICIT NONE
```

```
7.
8.     integer*4 :: i
9.     integer*4, parameter :: N=8
10.    real*4, Dimension(N) :: a, b
11.
12.    DO i=1,N
13.      a(i)=i*1.0
14.      b(i)=2.0
15.    END DO
16.
17.     print *, 'a = ', (a(i), i=1,N)
18.
19.      CALL kernel_wrapper(a, b, N)
20.
21.     print *, 'a + 2 = ', (a(i), i=1,N)
22.
23.    END PROGRAM
```


Makefile to compile the program fortest.f95

```
Test: fortest.f95 cudatest.o
        gfortran -L /usr/local/cuda/lib -I /usr/local/cuda/include -lcudart -lcuda fortest.f95 cudates
cudatest.o: cudatest.cu
        nvcc -c -O3 cudatest.cu
clean:
        rm a.out cudatest.o cudatest.linkinfo
```

3-3 Theano Programming

HPMC calculators have Theano already installed on them.

Theano is a Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. Theano combines the convenience of NumPy's syntax.

■ Theano Example

You can program in python and run theano on your HPMC.

python exthea1.py

```
07:10 qn001: /appz/theano # python exthea1.py
Using gpu device 0: Graphics Device
[GpuElemwise{exp,no_inplace}(<CudaNdarrayType(float32, vector)>), HostFromGpu(GpuElemwise{exp,no_inplace
Looping 1000 times took 0.429893016815 seconds
Result is [ 1.23178029  1.61879349  1.52278066 ...,  2.20771813  2.29967761
  1.62323296]
Used the gpu
```

python exthea2.py

```
07:11 qn001: /appz/theano # python exthea2.py
Using gpu device 0: Graphics Device
[GpuElemwise{exp,no_inplace}(<CudaNdarrayType(float32, vector)>)]
Looping 1000 times took 0.204324007034 seconds
Result is <CudaNdarray object at 0x6f05130>
Numpy result is [ 1.23178029  1.61879349  1.52278066 ...,  2.20771813  2.29967761
  1.62323296]
Used the gpu
```

python exthea3.py

```
07:11 qn001: /appz/theano # python exthea3.py
Using gpu device 0: Graphics Device
Used the gpu
target values for D
[ 1.  0.  1.  1.  0.  1.  1.  0.  0.  0.  0.  1.  0.  0.  1.  0.  0.  1.
  0.  0.  0.  1.  0.  0.  1.  0.  0.  0.  0.  0.  1.  1.  1.  1.  1.  0.
  0.  0.  0.  1.  1.  0.  1.  0.  0.  0.  1.  0.  0.  1.  1.  0.  0.  0.
  1.  0.  1.  1.  0.  1.  1.  1.  1.  0.  0.  1.  1.  1.  0.  1.  1.  0.
  1.  1.  1.  1.  0.  1.  0.  1.  1.  1.  1.  0.  0.  0.  1.  1.  1.  0.
  1.  0.  0.  0.  0.  1.  0.  0.  1.  1.  1.  1.  0.  1.  1.  1.  0.  0.
  0.  0.  1.  0.  0.  0.  1.  0.  0.  1.  1.  1.  1.  0.  0.  0.  1.  1.
  1.  0.  1.  0.  0.  0.  0.  0.  1.  0.  1.  0.  1.  1.  1.  0.  0.  0.
  1.  1.  0.  0.  0.  1.  1.  0.  0.  1.  1.  0.  1.  1.  0.  0.  0.  1.
  0.  1.  0.  1.  0.  0.  1.  0.  1.  0.  0.  1.  1.  1.  0.  1.  0.  0.
  1.  1.  0.  1.  0.  0.  0.  0.  1.  1.  1.  1.  0.  1.  1.  1.  0.  0.
  0.  1.  1.  1.  0.  0.  1.  1.  0.  1.  1.  0.  1.  1.  1.  1.  1.  0.
  0.  0.  0.  0.  0.  1.  0.  0.  1.  0.  0.  0.  0.  1.  0.  0.  0.  1.
  0.  1.  1.  0.  1.  1.  1.  1.  1.  1.  1.  1.  0.  1.  0.  0.  1.  1.
  1.  0.  1.  1.  0.  1.  0.  0.  0.  1.  0.  0.  1.  1.  0.  1.  0.  0.
  0.  1.  1.  1.  0.  1.  0.  0.  0.  1.  1.  1.  0.  0.  0.  1.  1.  1.
  0.  1.  0.  0.  0.  1.  1.  0.  0.  1.  1.  1.  0.  0.  1.  1.  1.  0.
  1.  0.  1.  1.  1.  1.  1.  0.  1.  0.  0.  1.  0.  0.  1.  1.  0.  1.
  1.  1.  1.  1.  1.  1.  0.  0.  0.  0.  0.  1.  1.  0.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  1.  0.  1.  1.  0.  1.  1.  1.  1.  0.  0.  0.  0.
  0.  0.  0.  1.  0.  1.  1.  1.  0.  1.  1.  0.  1.  0.  1.  1.  1.  1.
  1.  0.  0.  1.  1.  1.  0.  0.  0.  1.  0.  1.  0.  1.  0.  1.  1.  1.
  1.  1.  0.  0.]
```

prediction on D

```
[1 0 1 1 0 1 1 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 1 1 1 1 1 0 0
 0 0 1 1 0 1 0 0 0 1 0 0 1 1 0 0 0 1 0 1 1 0 1 1 1 1 0 0 1 1 1 0 1 1 0 1 1
 1 1 0 1 0 1 1 1 1 0 0 0 1 1 1 0 1 0 0 0 0 0 1 0 0 1 1 1 1 0 1 1 1 0 0 0 0 1
 0 0 0 1 0 0 1 1 1 1 0 0 0 1 1 1 0 1 0 0 0 0 0 0 1 0 1 0 1 1 1 0 0 0 1 1 0 0
 0 1 1 0 0 1 1 0 1 1 0 0 0 1 0 1 0 1 0 0 1 0 1 0 0 1 1 1 0 1 0 0 1 1 0 1 0
 0 0 0 1 1 1 1 1 0 1 1 1 0 0 0 1 1 1 0 0 1 1 0 1 1 0 1 1 1 1 1 1 0 0 0 0 0 0 1
 0 0 1 0 0 0 0 0 1 0 0 0 1 0 1 1 0 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 0 1 1 0 1 0
 0 0 1 0 0 1 1 0 1 0 0 0 1 1 1 0 1 0 0 0 1 1 1 0 0 0 1 1 1 0 1 0 0 0 1 1 0
 0 1 1 1 0 0 1 1 1 0 1 0 1 1 1 1 1 1 0 1 0 0 1 0 0 1 1 0 1 1 1 1 1 1 1 0 0 0
 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 0 0 0 0 0 0 0 1 0 1 1 1 0 1
 1 0 1 0 1 1 1 1 1 0 0 1 1 1 0 0 0 1 0 1 0 1 0 1 1 1 1 1 0 0]
```

Chapter 4
Configuring Infiniband for RDMA

Image File

The HPMC can be extended to access neighboring nodes memory via Infiniband appliances for RDMA access. This is realized through a switch connectivity that requires the following components:
The switch 12200:
Qlogic 12200-18 18-Port QDR InfiniBand Network Switch 12200-18-28
The cables:
Intel Qlogic CBL2-1001001-3, 40G QSFP+ Active Optical Cable QDR
QLOGIC CBL1-0600230 2M Cable Cop 30 AWG STD QSFP to QSFP
EMC 003-0080-01 3 Meter Infiniband 40gb QSFP to Cx4
EMC 003-0080-01 3 METER 40gb QSFP to Cx4 Cable

qle7340
QLogic QLE7340 1-Port 40Gb QSFP Infiniband QDR PCIe HCA Host Channel Adapter
QLogic InfiniPath QLE7140 IB6110401-01 InfiniBand SDR 4x 10Gb PCI-e x8

Image File

Read Chapter 4
Back to Contents

4-1 Configuring Infiniband for RDMA

The HPMC can be connected to multiple nodes over a switch.

The OFED packages required

```
ofed-scripts
rdma-core
rdma-core-devel
libibverbs
libibverbs-utils
librdmacm
librdmacm-utils
libibumad
ibacm
iwpmd
srp_daemon
mstflint
ofed-docs
compat-rdma
```

■ Probe The Infiniband HCA Cards

```
# lspci -nn | grep HCA
```

```
03:03 root@HPMC9: /mm03fs/MPI #  lspci -nn | grep HCA
5e:00.0 InfiniBand [0c06]: QLogic Corp. IBA7322 QDR InfiniBand HCA [1077:7322] (rev 02)
5f:00.0 InfiniBand [0c06]: QLogic Corp. IBA7322 QDR InfiniBand HCA [1077:7322] (rev 02)
```

```
#  grep -i 1fc1 /lib/modules/3.10.0-693.el7.x86_64/modules.*
```

```
03:07 root@HPMC9: /mm03fs/MPI #  grep -i 1fc1 /lib/modules/3.10.0-693.el7.x86_64/modules.*
/lib/modules/3.10.0-693.el7.x86_64/modules.alias:alias pci:v00001FC1d0000000Dsv*sd*bc*sc*i* ib_ipath
/lib/modules/3.10.0-693.el7.x86_64/modules.alias:alias pci:v00001FC1d00000010sv*sd*bc*sc*i* ib_qib
```

■ Adding the Infiniband to the Network Services

Edit the ifcfg-ib0

```
# vi /etc/sysconfig/network-scripts/ifcfg-ib0
```

```
NAME=ib0
DEVICE=ib0
TYPE=InfiniBand
# to check if connected, cat /sys/class/net/ib0/mode
CONNECTED_MODE=no
PROXY_METHOD=none
BOOTPROTO=static
STARTMODE=auto
NM_CONTROLLED=no
# NOTE if subnet then possible with 192.168.1.X as
#    PADDR=192.168.1.119 NETWORK=192.168.1.0 BROADCAST=192.168.1.255
#    also make changes in  ifcfg-ib0.8002
IPADDR=172.16.0.132
PREFIX=16
NETMASK=255.255.0.0
NETWORK=172.16.0.0
BROADCAST=172.16.255.255
#
ONBOOT=yes
DEFROUTE=yes
MTU=65520
HWADDR=80:00:00:03:FE:80:00:00:00:00:00:00:00:00:11:75:00:00:6E:EF:26
```

■ Check Infiniband Modules and Services

Check the Infiniband related modules:

```
# lsmod | grep ib
```

```
03:12 root@HPMC9: /mm03fs/MPI #  lsmod | grep ib
ib_ucm                 22589  0
ib_uverbs              64636  16 ib_ucm,rdma_ucm
ib_iser                47813  0
rdma_cm                54426  3 rpcrdma,ib_iser,rdma_ucm
ib_umad                22080  8
libiscsi               57233  1 ib_iser
ib_ipoib              110142  0
scsi_transport_iscsi   99909  2 ib_iser,libiscsi
ib_cm                  47287  3 rdma_cm,ib_ucm,ib_ipoib
ib_qib                362592  4
rdmavt                 59235  1 ib_qib
ib_core               211874  15 rdma_cm,i40iw,ib_cm,iw_cm,rpcrdma,ib_qib,ib_ucm,rdmavt,ib_iser,ib_umad,ib_uverbs,r
dca                    15130  1 ib_qib
libnvdimm             132047  1 nfit
libahci                31992  1 ahci
libata                238896  2 ahci,libahci
```

Check the opensm service to be up:

systemctl list-unit-files | grep opensm

```
03:07 root@HPMC9 #  systemctl list-unit-files    | grep open
opensm.service                              enabled
```

■ Query the Infiniband Connectivity

ibnodes

```
03:14 root@HPMC9: /mm03fs/MPI #  ibnodes
Ca      : 0x00117500006ef940 ports 1 "HPMC7 qib0"
Ca      : 0x0011750000794fd8 ports 1 "HPMC9 qib1"
Ca      : 0x00117500006eef26 ports 1 "HPMC9 qib0"
Switch  : 0x00066a00e3004e75 ports 36 "QLogic 12200-18 GUID=0x00066a00e3004e75" base port 0 lid 4 lmc 0
```

ibv_devices

```
03:18 root@HPMC9: /mm03fs/MPI #  ibv_devices
    device                 node GUID
    ------              ----------------
    qib0                00117500006eef26
    qib1                0011750000794fd8
    i40iw0              002590bc3ab90000
    i40iw1              002590bc3ab80000
```

ibnodes

```
03:18 root@HPMC9: /mm03fs/MPI #  ibnodes
Ca      : 0x00117500006ef940 ports 1 "HPMC7 qib0"
Ca      : 0x0011750000794fd8 ports 1 "HPMC9 qib1"
Ca      : 0x00117500006eef26 ports 1 "HPMC9 qib0"
Switch  : 0x00066a00e3004e75 ports 36 "QLogic 12200-18 GUID=0x00066a00e3004e75" base port 0 lid 4 lmc 0
<pre>


((#))  ibnetdiscover
<pre>
03:18 root@HPMC9: /mm03fs/MPI #  ibnetdiscover
#
# Topology file: generated on Fri Apr 17 03:18:50 2026
#
# Initiated from node 00117500006eef26 port 00117500006eef26

vendid=0x66a
devid=0x7320
sysimgguid=0x66a00e3004e75
switchguid=0x66a00e3004e75(66a00e3004e75)
Switch  36 "S-00066a00e3004e75"         # "QLogic 12200-18 GUID=0x00066a00e3004e75" base port 0 lid 4 lmc 0
[10]    "H-00117500006eef26"[1](117500006eef26)                 # "HPMC9 qib0" lid 2 4xQDR
[11]    "H-00117500006ef940"[1](117500006ef940)                 # "HPMC7 qib0" lid 5 4xDDR
[12]    "H-0011750000794fd8"[1](11750000794fd8)                 # "HPMC9 qib1" lid 1 4xQDR

vendid=0x1175
devid=0x7322
sysimgguid=0x117500006ef940
caguid=0x117500006ef940
Ca      1 "H-00117500006ef940"          # "HPMC7 qib0"
[1](117500006ef940)     "S-00066a00e3004e75"[11]                # lid 5 lmc 0 "QLogic 12200-18 GUID=0x00066a00e3004

vendid=0x1175
devid=0x7322
```

```
sysimgguid=0x117500006eef26
caguid=0x11750000794fd8
Ca      1 "H-0011750000794fd8"          # "HPMC9 qib1"
[1](11750000794fd8)     "S-00066a00e3004e75"[12]                # lid 1 lmc 0 "QLogic 12200-18 GUID=0x00066a00e3004

vendid=0x1175
devid=0x7322
sysimgguid=0x117500006eef26
caguid=0x117500006eef26
Ca      1 "H-00117500006eef26"          # "HPMC9 qib0"
[1](117500006eef26)     "S-00066a00e3004e75"[10]                #
```

■ Query the Infiniband Devices Info

ibv_devinfo -d qib0 -v

```
03:18 root@HPMC9: /mm03fs/MPI #  ibv_devinfo -d qib0 -v
hca_id: qib0
        transport:                      InfiniBand (0)
        fw_ver:                         0.0.0
        node_guid:                      0011:7500:006e:ef26
        sys_image_guid:                 0011:7500:006e:ef26
        vendor_id:                      0x1175
        vendor_part_id:                 29474
        hw_ver:                         0x2
        board_id:                       InfiniPath_QLE7340
        phys_port_cnt:                  1
        max_mr_size:                    0xffffffffffffffff
        page_size_cap:                  0x1000
        max_qp:                         16384
        max_qp_wr:                      16383
        device_cap_flags:               0x00003d06
                                        BAD_PKEY_CNTR
                                        BAD_QKEY_CNTR
                                        SHUTDOWN_PORT
                                        PORT_ACTIVE_EVENT
                                        SYS_IMAGE_GUID
                                        RC_RNR_NAK_GEN
                                        SRQ_RESIZE
        max_sge:                        96
        max_sge_rd:                     96
        max_cq:                         131071
        max_cqe:                        196607
        max_mr:                         0
        max_pd:                         65535
        max_qp_rd_atom:                 16
        max_ee_rd_atom:                 0
        max_res_rd_atom:                0
        max_qp_init_rd_atom:            255
        max_ee_init_rd_atom:            0
        atomic_cap:                     ATOMIC_GLOB (2)
        max_ee:                         0
        max_rdd:                        0
        max_mw:                         0
        max_raw_ipv6_qp:                0
        max_raw_ethy_qp:                0
        max_mcast_grp:                  16384
        max_mcast_qp_attach:            16
        max_total_mcast_qp_attach:      262144
        max_ah:                         65535
        max_fmr:                        0
        max_srq:                        1024
        max_srq_wr:                     131071
        max_srq_sge:                    128
        max_pkeys:                      4
        local_ca_ack_delay:             0
        general_odp_caps:
        rc_odp_caps:
                                        NO SUPPORT
        uc_odp_caps:
                                        NO SUPPORT
        ud_odp_caps:
                                        NO SUPPORT
        completion_timestamp_mask not supported
        core clock not supported
        device_cap_flags_ex:            0x0
        tso_caps:
        max_tso:                        0
        rss_caps:
                max_rwq_indirection_tables:                     0
                max_rwq_indirection_table_size:                 0
                rx_hash_function:                               0x0
                rx_hash_fields_mask:                            0x0
        max_wq_type_rq:                 0
        packet_pacing_caps:
                qp_rate_limit_min:      0kbps
                qp_rate_limit_max:      0kbps
        tag matching not supported
                port:   1
                        state:                  PORT_ACTIVE (4)
                        max_mtu:                4096 (5)
                        active_mtu:             2048 (4)
                        sm_lid:                 2
                        port_lid:               2
                        port_lmc:               0x00
                        link_layer:             InfiniBand
```

```
                        max_msg_sz:             0x80000000
                        port_cap_flags:         0x0761086a
                        max_vl_num:             2 (2)
                        bad_pkey_cntr:          0x0
                        qkey_viol_cntr:         0x0
                        sm_sl:                  0
                        pkey_tbl_len:           4
                        gid_tbl_len:            5
                        subnet_timeout:         18
                        init_type_reply:        0
                        active_width:           4X (2)
                        active_speed:           10.0 Gbps (4)
                        phys_state:             LINK_UP (5)
                        GID[  0]:               fe80:0000:0000:0000:0011:7500:006e:ef26
```

ibv_devinfo -d qib1 -v

```
03:22 root@HPMC9: /mm03fs/MPI #  ibv_devinfo -d qib1 -v
hca_id: qib1
        transport:                      InfiniBand (0)
        fw_ver:                         0.0.0
        node_guid:                      0011:7500:0079:4fd8
        sys_image_guid:                 0011:7500:006e:ef26
        vendor_id:                      0x1175
        vendor_part_id:                 29474
        hw_ver:                         0x2
        board_id:                       InfiniPath_QLE7340
        phys_port_cnt:                  1
        max_mr_size:                    0xffffffffffffffff
        page_size_cap:                  0x1000
        max_qp:                         16384
        max_qp_wr:                      16383
        device_cap_flags:               0x00003d06
                                        BAD_PKEY_CNTR
                                        BAD_QKEY_CNTR
                                        SHUTDOWN_PORT
                                        PORT_ACTIVE_EVENT
                                        SYS_IMAGE_GUID
                                        RC_RNR_NAK_GEN
                                        SRQ_RESIZE
        max_sge:                        96
        max_sge_rd:                     96
        max_cq:                         131071
        max_cqe:                        196607
        max_mr:                         0
        max_pd:                         65535
        max_qp_rd_atom:                 16
        max_ee_rd_atom:                 0
        max_res_rd_atom:                0
        max_qp_init_rd_atom:            255
        max_ee_init_rd_atom:            0
        atomic_cap:                     ATOMIC_GLOB (2)
        max_ee:                         0
        max_rdd:                        0
        max_mw:                         0
        max_raw_ipv6_qp:                0
        max_raw_ethy_qp:                0
        max_mcast_grp:                  16384
        max_mcast_qp_attach:            16
        max_total_mcast_qp_attach:      262144
        max_ah:                         65535
        max_fmr:                        0
        max_srq:                        1024
        max_srq_wr:                     131071
        max_srq_sge:                    128
        max_pkeys:                      4
        local_ca_ack_delay:             0
        general_odp_caps:
        rc_odp_caps:
                                        NO SUPPORT
        uc_odp_caps:
                                        NO SUPPORT
        ud_odp_caps:
                                        NO SUPPORT
        completion_timestamp_mask not supported
        core clock not supported
        device_cap_flags_ex:            0x0
        tso_caps:
        max_tso:                        0
        rss_caps:
                max_rwq_indirection_tables:                     0
                max_rwq_indirection_table_size:                 0
                rx_hash_function:                               0x0
                rx_hash_fields_mask:                            0x0
        max_wq_type_rq:                 0
        packet_pacing_caps:
                qp_rate_limit_min:      0kbps
                qp_rate_limit_max:      0kbps
        tag matching not supported
                port:   1
                        state:                  PORT_ACTIVE (4)
                        max_mtu:                4096 (5)
                        active_mtu:             2048 (4)
                        sm_lid:                 2
                        port_lid:               1
                        port_lmc:               0x00
                        link_layer:             InfiniBand
                        max_msg_sz:             0x80000000
                        port_cap_flags:         0x07610868
```

```
max_vl_num:             2 (2)
bad_pkey_cntr:          0x0
qkey_viol_cntr:         0x0
sm_sl:                  0
pkey_tbl_len:           4
gid_tbl_len:            5
subnet_timeout:         18
init_type_reply:        0
active_width:           4X (2)
active_speed:           10.0 Gbps (4)
phys_state:             LINK_UP (5)
GID[  0]:               fe80:0000:0000:0000:0011:7500:0079:4fd8
```

■ Display the Network Interface Parameter

ifconfig

```
br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.131  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::225:90ff:febc:3ab8  prefixlen 64  scopeid 0x20<link>
        ether 00:25:90:bc:3a:b8  txqueuelen 1000  (Ethernet)
        RX packets 717476  bytes 900282208 (858.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 371423  bytes 886594089 (845.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::225:90ff:febc:3ab8  prefixlen 64  scopeid 0x20<link>
        ether 00:25:90:bc:3a:b8  txqueuelen 1000  (Ethernet)
        RX packets 719099  bytes 910433990 (868.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 948123  bytes 924656789 (881.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eno2: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 00:25:90:bc:3a:b9  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ib0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 2044
        inet 172.16.0.132  netmask 255.255.0.0  broadcast 172.16.255.255
        inet6 fe80::211:7500:6e:ef26  prefixlen 64  scopeid 0x20<link>
Infiniband hardware address can be incorrect! Please read BUGS section in ifconfig(8).
        infiniband 80:00:00:03:FE:80:00:00:00:00:00:00:00:00:00:00:00:00:00:00  txqueuelen 256  (InfiniBand)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ib1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 65520
        inet 172.16.0.133  netmask 255.255.0.0  broadcast 172.16.255.255
        inet6 fe80::211:7500:79:4fd8  prefixlen 64  scopeid 0x20<link>
Infiniband hardware address can be incorrect! Please read BUGS section in ifconfig(8).
        infiniband 80:00:00:03:FE:80:00:00:00:00:00:00:00:00:00:00:00:00:00:00  txqueuelen 256  (InfiniBand)
        RX packets 6  bytes 336 (336.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ib0.8002: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 2044
        inet6 fe80::211:7500:6e:ef26  prefixlen 64  scopeid 0x20<link>
Infiniband hardware address can be incorrect! Please read BUGS section in ifconfig(8).
        infiniband 80:00:00:07:FE:80:00:00:00:00:00:00:00:00:00:00:00:00:00:00  txqueuelen 256  (InfiniBand)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ib1.8002: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 65520
        inet 172.16.0.133  netmask 255.255.0.0  broadcast 172.16.255.255
        inet6 fe80::211:7500:79:4fd8  prefixlen 64  scopeid 0x20<link>
Infiniband hardware address can be incorrect! Please read BUGS section in ifconfig(8).
        infiniband 80:00:00:07:FE:80:00:00:00:00:00:00:00:00:00:00:00:00:00:00  txqueuelen 256  (InfiniBand)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 80871  bytes 864928828 (824.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 80871  bytes 864928828 (824.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mic0: flags=67<UP,BROADCAST,RUNNING>  mtu 64512
        ether 0a:c9:28:ae:ba:9f  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Chapter 5
RDMA Over Infiniband Connectivity

Image File

Remote Direct Memory Access (RDMA) is a high-performance networking technology that allows a device to transfer data directly to or from the memory of another device, bypassing both hosts' operating systems, CPUs, and caches. This results in ultra-low latency, high throughput, and lower CPU overhead, making it essential for AI/ML training, HPC, and fast storage, often using InfiniBand or RoCE.

RoCE (RDMA over Converged Ethernet) and InfiniBand (IB) are both high-performance, low-latency networking technologies utilizing RDMA to bypass CPU kernels, primarily used in AI and HPC clusters. InfiniBand offers superior performance and a native lossless, simplified architecture, while RoCE provides cost-effective scalability, leveraging existing Ethernet infrastructure.

Image File

Read Chapter 5
Back to Contents

5-1 RDMA Over Infiniband Connectivity

HPMC calculators that are fabric capable can be interconnected together to take advange of RDMA via DAPL to execute programs and distribute the processing on neighboring HPMC's.

When the HPMC is equipped with a switching card device named qib0, it is capable to communicate with a neighboring HPMC that equipped with a similar card via a switch.

To find out if your HPMC is RDMA capable and it is running properly, you need to verify that "IB/RoCE v1" is up. The following steps show how to determine if your HPMC is "IB/RoCE v1" capable.

Make sure opensm is running:

systemctl status opensm

Make sure DAPL is running:

Make sure RoCE is running:

To validate that RoCE is "IB/RoCE v1" up:

mkdir /sys/kernel/config/rdma_cm/qib0

tree /sys/kernel/config/rdma_cm/qib0

cat /sys/kernel/config/rdma_cm/qib0/ports/1/default_roce_mode

The above command should print: **"IB/RoCE v1"**

```
#  mkdir /sys/kernel/config/rdma_cm/qib0
#  tree /sys/kernel/config/rdma_cm/qib0
/sys/kernel/config/rdma_cm/qib0
â””â”€â”€ ports
    â””â”€â”€ 1
        â””â”€â”€ default_roce_mode

2 directories, 1 file
#  cat /sys/kernel/config/rdma_cm/qib0/ports/1/default_roce_mode
IB/RoCE v1
```

The device qib0 is configured on the network as ib0. Any of the following commands can be used to query ib0:

ip link show ib0

ifconfig ib0

ibv_devinfo -v qib0

```
#  ip link show ib0
5: ib0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 2044 qdisc pfifo_fast state UP mode DEFAULT group default qlen 256
    link/infiniband 80:00:00:03:fe:80:00:00:00:00:00:00:00:11:75:00:00:6e:ef:26 brd 00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:
#  ifconfig ib0
ib0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 2044
        inet 172.16.0.132  netmask 255.255.0.0  broadcast 172.16.255.255
        inet6 fe80::211:7500:6e:ef26  prefixlen 64  scopeid 0x20<link>
Infiniband hardware address can be incorrect! Please read BUGS section in ifconfig(8).
        infiniband 80:00:00:03:FE:80:00:00:00:00:00:00:00:00:00:00:00:00:00:00  txqueuelen 256  (InfiniBand)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

#  ibv_devinfo -v qib0
hca_id: qib0
        transport:                      InfiniBand (0)
        fw_ver:                         0.0.0
        node_guid:                      0011:7500:006e:ef26
        sys_image_guid:                 0011:7500:006e:ef26
        vendor_id:                      0x1175
...
```

Make sure that it is set to **datagram** mode (this configuration is set in the network scripts: ifcnf-ib0 ifcnf-ib0.8002

cat /sys/class/net/ib0/mode

To change the mode to **connected**, edit the ifcnf-ib0 ifcnf-ib0.8002 and reboot your HPMC.

HPMC calculators branded as HPMC-KLIB, KLIB Knight-Landing InfiniBand, are all fabric connectivity ready and can be plugged to the switch.

In the following chapter we will show some of the commands to test and measure the performance of RDMA between the nodes.

■ Testing Connectivity with ping

Testing the connectivity with pingpong between hpmc9 and hpmc7 that are interconnected by infiniband.

16:19 root@HPMC9: /mm03fs/MPI # ibv_rc_pingpong -d qib0 -g 0 -i 1

Start ibping as a server on hpmc9

16:19 root@HPMC9: /mm03fs/MPI # ibping -S -P 1 -C qib0

From hpmc7 ping hpmc9 via qib0 Lid 2

16:06 root@HPMC7: ~ # ibping -c 10000 -f -C qib0 -P 1 -G 0x00117500006eef26

```
16:06 root@HPMC7: ~ #   ibping -c 10000 -f -C qib0 -P 1 -G 0x00117500006eef26

---  (Lid 2) ibping statistics ---
10000 packets transmitted, 0 received, 100% packet loss, time 574 ms
rtt min/avg/max = 0.000/0.000/0.000 ms
```

■ Testing Connectivity with Iterations Using pingpong

Testing the connectivity with pingpong between hpmc9 and hpmc7 that are interconnected by infiniband.

16:19 root@HPMC9: /mm03fs/MPI # ibv_rc_pingpong -d qib0 -g 0 -i 1

```
16:19 root@HPMC9: /mm03fs/MPI #  ibv_rc_pingpong -d qib0 -g 0 -i 1
  local address:  LID 0x0002, QPN 0x00001b, PSN 0xadb290, GID fe80::11:7500:6e:ef26
  remote address: LID 0x0005, QPN 0x00001b, PSN 0x0c5add, GID fe80::11:7500:6e:f940
8192000 bytes in 0.04 seconds = 1464.59 Mbit/sec
1000 iters in 0.04 seconds = 44.75 usec/iter
```

15:58 root@HPMC7: ~ # ibv_rc_pingpong -g 0 -d qib0 -i 1 192.168.0.131

```
15:58 root@HPMC7: ~ #  ibv_rc_pingpong -g 0 -d qib0 -i 1 192.168.0.131
  local address:  LID 0x0005, QPN 0x00001b, PSN 0x0c5add, GID fe80::11:7500:6e:f940
  remote address: LID 0x0002, QPN 0x00001b, PSN 0xadb290, GID fe80::11:7500:6e:ef26
8192000 bytes in 0.04 seconds = 1478.27 Mbit/sec
1000 iters in 0.04 seconds = 44.33 usec/iter
```

■ RDMA WRITE_BW

Both hpmc9 and hpmc7 are interconnected by infiniband.

16:27 root@HPMC9: /mm03fs/MPI # ib_write_bw -R

```
16:27 root@HPMC9: /mm03fs/MPI #  ib_write_bw -R
************************************
* Waiting for client to connect... *
************************************
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : qib0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 CQ Moderation   : 100
 Mtu             : 2048[B]
 Link type       : IB
 Max inline data : 0[B]
 rdma_cm QPs     : ON
 Data ex. method : rdma_cm
---------------------------------------------------------------------------------------
 Waiting for client rdma_cm QP to connect
 Please run the same command with the IB/RoCE interface IP
---------------------------------------------------------------------------------------
 local address: LID 0x02 QPN 0x001f PSN 0x3c0bc5
 remote address: LID 0x05 QPN 0x001f PSN 0x280903
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]
 65536      5000             628.34             618.89             0.009902
---------------------------------------------------------------------------------------
```

16:06 root@HPMC7: ~ # ib_write_bw hpmc9ib0 -R --cpu_util

```
16:06 root@HPMC7: ~ #  ib_write_bw hpmc9ib0 -R --cpu_util
---------------------------------------------------------------------------------------
 CPU Utilization works only with Duration mode.
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : qib0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 TX depth        : 128
 CQ Moderation   : 100
 Mtu             : 2048[B]
 Link type       : IB
```

```
 Max inline data : 0[B]
 rdma_cm QPs     : ON
 Data ex. method : rdma_cm
---------------------------------------------------------------------------------------
 local address: LID 0x05 QPN 0x001f PSN 0x280903
 remote address: LID 0x02 QPN 0x001f PSN 0x3c0bc5
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]
Conflicting CPU frequency values detected: 1501.062000 != 1096.101000. CPU Frequency is not max.
 65536      5000             628.34             618.89             0.009902
---------------------------------------------------------------------------------------
```

■ RDMA WRITE CPU_UTIL

Both hpmc9 and hpmc7 are interconnected by infiniband.

00:58 root@HPMC9: ~ # ib_write_bw -F -c RC --cpu_util -D 10

```
00:58 root@HPMC9: ~ #  ib_write_bw -F  -c RC --cpu_util -D 10

************************************
* Waiting for client to connect... *
************************************
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : qib0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 CQ Moderation   : 100
 Mtu             : 2048[B]
 Link type       : IB
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0x01 QPN 0x09d4 PSN 0x2196eb RKey 0x2505100 VAddr 0x007f84a816a000
 remote address: LID 0x05 QPN 0x0019 PSN 0x6d2660 RKey 0x0a0b00 VAddr 0x007f296b6c3000
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]    CPU_Util[%]
 65536      57900            0.00               5.06               0.009648         0.00
---------------------------------------------------------------------------------------
```

00:50 root@HPMC7: ~ # ib_write_bw -c RC --report_gbits -F -d qib0 hpmc9 -c RC --cpu_util -D 10

```
00:50 root@HPMC7: ~ #  ib_write_bw -c RC --report_gbits  -F -d qib0 hpmc9 -c RC  --cpu_util -D 10
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : qib0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 TX depth        : 128
 CQ Moderation   : 100
 Mtu             : 2048[B]
 Link type       : IB
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0x05 QPN 0x0019 PSN 0x6d2660 RKey 0x0a0b00 VAddr 0x007f296b6c3000
 remote address: LID 0x01 QPN 0x09d4 PSN 0x2196eb RKey 0x2505100 VAddr 0x007f84a816a000
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[Gb/sec]    BW average[Gb/sec]   MsgRate[Mpps]    CPU_Util[%]
 65536      57900            0.00               5.06               0.009648         0.36
---------------------------------------------------------------------------------------
```

■ RDMA WRITE LATENCY

Both hpmc9 and hpmc7 are interconnected by infiniband.

01:23 root@HPMC9: # ib_write_lat -a -d qib0 -i 1 --report_gbits -F -n 1000

```
01:23 root@HPMC9: ~ #  ib_write_lat -a -d qib0 -i 1 --report_gbits -F -n 1000

************************************
* Waiting for client to connect... *
************************************
---------------------------------------------------------------------------------------
                    RDMA_Write Latency Test
 Dual-port       : OFF          Device         : qib0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 Mtu             : 2048[B]
 Link type       : IB
 Max inline data : 0[B]
```

```
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0x01 QPN 0x09ec PSN 0x7ddf72 RKey 0x2646500 VAddr 0x007fa7d18c3000
 remote address: LID 0x05 QPN 0x0029 PSN 0xc1d796 RKey 0x1e1f00 VAddr 0x007ffbde44c000
---------------------------------------------------------------------------------------
 #bytes #iterations    t_min[usec]    t_max[usec]  t_typical[usec]    t_avg[usec]    t_stdev[usec]   99% percentile[usec]
 2       1000          6.23           57.38        7.30               7.48           3.09            19.36
 4       1000          6.20           41.64        6.62               7.34           2.32            17.11
 8       1000          5.85           973.74       7.27               7.51           2.88            21.10
 16      1000          5.93           41.00        7.28               7.51           2.44            17.73
 32      1000          6.20           40.25        7.29               7.53           2.39            18.21
 64      1000          6.24           38.57        7.36               7.66           2.59            19.52
 128     1000          6.33           44.93        7.45               7.86           2.46            18.67
 256     1000          6.33           43.53        7.41               7.91           2.27            16.97
 512     1000          6.73           43.11        7.46               8.10           2.29            16.52
 1024    1000          7.32           42.31        8.82               8.70           2.24            18.12
 2048    1000          7.68           43.44        9.01               9.38           2.18            17.30
 4096    1000          8.92           40.99        9.73               10.19          2.29            21.20
 8192    1000          11.80          65.74        12.68              13.15          2.37            21.79
 16384   1000          17.30          52.91        18.97              19.37          2.39            27.93
 32768   1000          27.50          64.84        30.05              30.48          2.61            39.86
 65536   1000          46.11          98.02        51.34              51.55          3.49            69.52
 131072  1000          84.70          181.23       91.18              94.01          11.50           158.25
 262144  1000          160.20         318.40       164.29             169.40         20.16           302.77
 524288  1000          315.31         631.62       481.23             491.63         48.41           618.88
 1048576 1000          668.80         1892.48      980.28             975.48         97.46           1184.61
 2097152 1000          1581.80        2357.73      1999.08            2001.44        159.81          2335.71
 4194304 1000          2980.43        6388.11      4170.43            4265.46        455.42          5018.32
 8388608 1000          6790.65        19943.24      9643.13           9673.86        460.42          11191.81
---------------------------------------------------------------------------------------
```

01:15 root@HPMC7: ~ # ib_write_lat -n 1000 -a -F hpmc9ib0

```
01:15 root@HPMC7: ~ #  ib_write_lat -n 1000 -a -F hpmc9ib0
---------------------------------------------------------------------------------------
                    RDMA_Write Latency Test
 Dual-port       : OFF          Device         : qib0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 TX depth        : 1
 Mtu             : 2048[B]
 Link type       : IB
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0x05 QPN 0x0029 PSN 0xc1d796 RKey 0x1e1f00 VAddr 0x007ffbde44c000
 remote address: LID 0x01 QPN 0x09ec PSN 0x7ddf72 RKey 0x2646500 VAddr 0x007fa7d18c3000
---------------------------------------------------------------------------------------
 #bytes #iterations    t_min[usec]    t_max[usec]  t_typical[usec]    t_avg[usec]    t_stdev[usec]   99% percentile[usec]
 2       1000          6.26           65.92        7.11               7.45           2.72            18.53
 4       1000          6.30           35.33        6.85               7.32           2.00            17.23
 8       1000          6.22           77.72        7.12               7.48           2.50            17.39
 16      1000          6.26           38.88        7.19               7.48           2.22            16.28
 32      1000          6.25           36.02        7.17               7.51           2.10            17.28
 64      1000          6.31           40.74        7.30               7.63           2.28            17.50
 128     1000          6.34           34.77        7.45               7.84           2.19            17.62
 256     1000          6.46           37.31        7.43               7.88           1.99            17.07
 512     1000          6.78           32.63        7.49               8.08           2.05            17.07
 1024    1000          7.31           33.82        8.74               8.68           1.99            17.88
 2048    1000          7.72           38.05        9.01               9.35           1.91            17.68
 4096    1000          9.06           39.97        9.74               10.17          2.01            21.05
 8192    1000          11.98          66.97        12.72              13.12          2.17            20.83
 16384   1000          17.99          53.98        18.95              19.34          2.08            27.43
 32768   1000          28.76          63.88        30.04              30.46          2.39            37.89
 65536   1000          45.77          92.29        51.34              51.53          2.79            64.58
 131072  1000          87.75          172.78       91.35              93.99          10.07           151.73
 262144  1000          160.01         316.95       165.04             169.29         19.55           298.54
 524288  1000          314.70         627.75       481.06             491.46         47.88           617.67
 1048576 1000          677.13         1205.93      980.28             975.31         97.09           1184.69
 2097152 1000          1577.45        2366.46      1997.71            2001.18        159.95          2336.93
 4194304 1000          2990.30        5675.28      4171.74            4264.63        453.91          4979.38
 8388608 1000          6789.65        18944.48      9641.62           9664.59        360.09          11280.70
---------------------------------------------------------------------------------------
```

■ Measuring Performance with iperf3 Over Infiniband

Both hpmc9 and hpmc7 are interconnected by infiniband, we want to measure performace between the two nodes for sender and receiver.

01:32 root@HPMC9: ~ # iperf3 -i 5 -s -B hpmc9ib0

```
01:32 root@HPMC9: ~ #  iperf3 -i 5 -s -B hpmc9ib0
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 172.16.0.119, port 55015
[  5] local 172.16.0.132 port 5201 connected to 172.16.0.119 port 50086
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-5.00   sec  1.15 GBytes  1.98 Gbits/sec
[  5]   5.00-10.00  sec  1.18 GBytes  2.03 Gbits/sec
[  5]  10.00-10.04  sec  9.00 MBytes  2.02 Gbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.04  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-10.04  sec  2.34 GBytes  2.00 Gbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

```
Accepted connection from 172.16.0.119, port 34576
[  5] local 172.16.0.132 port 5201 connected to 172.16.0.119 port 34578
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-5.00   sec  1.13 GBytes  1.94 Gbits/sec
[  5]   5.00-10.00  sec  1.15 GBytes  1.97 Gbits/sec
[  5]  10.00-10.04  sec  8.31 MBytes  1.72 Gbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.04  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-10.04  sec  2.28 GBytes  1.95 Gbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 172.16.0.119, port 34580
[  5] local 172.16.0.132 port 5201 connected to 172.16.0.119 port 34582
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-5.00   sec  1.33 GBytes  2.29 Gbits/sec
[  5]   5.00-10.00  sec  1.51 GBytes  2.60 Gbits/sec
[  5]  10.00-10.04  sec  12.2 MBytes  2.56 Gbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.04  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-10.04  sec  2.86 GBytes  2.45 Gbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

On hpmc7 we run iperf3:

```
# 01:24 root@HPMC7: ~ # iperf3 -i 5 -B hpmc7ib0 -c hpmc9ib0

01:24 root@HPMC7: ~ #  iperf3 -i 5 -B hpmc7ib0 -c hpmc9ib0
Connecting to host hpmc9ib0, port 5201
[  4] local 172.16.0.119 port 50086 connected to 172.16.0.132 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-5.00   sec  1.16 GBytes  2.00 Gbits/sec    0    177 KBytes
[  4]   5.00-10.00  sec  1.18 GBytes  2.03 Gbits/sec    0    278 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  2.34 GBytes  2.01 Gbits/sec    0             sender
[  4]   0.00-10.00  sec  2.34 GBytes  2.01 Gbits/sec                  receiver

iperf Done.
01:25 root@HPMC7: ~ #  iperf3 -i 5 -c hpmc9ib0
Connecting to host hpmc9ib0, port 5201
[  4] local 172.16.0.119 port 34578 connected to 172.16.0.132 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-5.00   sec  1.14 GBytes  1.95 Gbits/sec    0    152 KBytes
[  4]   5.00-10.00  sec  1.15 GBytes  1.97 Gbits/sec    0    267 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  2.28 GBytes  1.96 Gbits/sec    0             sender
[  4]   0.00-10.00  sec  2.28 GBytes  1.96 Gbits/sec                  receiver

iperf Done.
01:26 root@HPMC7: ~ #  iperf3 -i 5 -c hpmc9ib0
Connecting to host hpmc9ib0, port 5201
[  4] local 172.16.0.119 port 34582 connected to 172.16.0.132 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-5.00   sec  1.35 GBytes  2.31 Gbits/sec    0    148 KBytes
[  4]   5.00-10.00  sec  1.51 GBytes  2.60 Gbits/sec    0    231 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  2.86 GBytes  2.45 Gbits/sec    0             sender
[  4]   0.00-10.00  sec  2.86 GBytes  2.45 Gbits/sec                  receiver

iperf Done.
```

■ Measuring Performance with iperf3 Over Ethernet

Here we measure performace between the nodes by binding to the ethernet. Notice how the transfer amd bandwith have dropped drappatically.

```
# 01:30 root@HPMC9: ~ # iperf3 -i 5 -s -B hpmc9

01:30 root@HPMC9: ~ #  iperf3 -i 5 -s -B hpmc9
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 192.168.0.117, port 56832
[  5] local 192.168.0.131 port 5201 connected to 192.168.0.117 port 56834
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-5.00   sec  55.6 MBytes  93.2 Mbits/sec
[  5]   5.00-10.00  sec  56.1 MBytes  94.0 Mbits/sec
[  5]  10.00-10.03  sec   396 KBytes  93.9 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.03  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-10.03  sec   112 MBytes  93.6 Mbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

```
# 01:24 root@HPMC7: ~ # iperf3 -i 5 -c hpmc9

01:24 root@HPMC7: ~ #  iperf3 -i 5 -c hpmc9
Connecting to host hpmc9, port 5201
[  4] local 192.168.0.117 port 56834 connected to 192.168.0.131 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
```

```
[  4]   0.00-5.00   sec  56.7 MBytes  95.2 Mbits/sec    0    194 KBytes
[  4]   5.00-10.00  sec  56.7 MBytes  95.1 Mbits/sec    0    260 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   113 MBytes  95.1 Mbits/sec    0             sender
[  4]   0.00-10.00  sec   112 MBytes  93.9 Mbits/sec                  receiver

iperf Done.
```

5-2 Fabric MPIRUN Offloading to MIC

HPMC calculators that are fabric capable can be interconnected together to take advange of RDMA via DAPL to execute programs and distribute the processing on neighboring HPMC's.

It is possible to interconnect two or more HPMC's DMA using switching networking cards, called infinibands. An application that is executed on the MIC of an HPMC can also be executed on the MAC of another HPMC system.

The core (* not CPU core) of such execution is laid through RDMA basic and essential components, all known to be stacked in what is called rdma-core. Message passing across the RDMA is essentially what allow the software application to be executed on a group (* we will avoid using cluster or clustering) of HPMC. The Message Passing Interface (MPI) is a set of RFC protocol (open or not open) that allow the execution of a program to be distributed on many HPMCs. While MPI is a monotonic interface that explore the RDMA and IPoIB switching (subject of DAPL) its execution benefit from leaving the Central Processing Unit (CPU) alone, hence it is asynchronous and fully duplexed depending on the netfiniend.

Consider the following trap_offload.c program that we will be running to test the MPI execution on our HPMCs.

-- Program Code 5.2.1 : [LISTING trap_offload.c] - [Trap Offload Program]

(raw text)

```
1.    #include <math.h>
2.    #include <mpi.h>
3.    #include <stdio.h>
4.    #include <unistd.h>
5.    #define NUM_TRAPEZOIDS 10000000000
6.    __attribute__((target(mic))) inline double f(double x) {
7.        return 1.00*x*x*exp(-(x-0.0)*(x-0.0)/(2.0*0.25*0.25))
8.            + 0.50*x*x*exp(-(x-0.2)*(x-0.2)/(2.0*0.50*0.50))
9.            + 0.50*x*x*exp(-(x+0.2)*(x+0.2)/(2.0*0.50*0.50))
10.           + 0.25*x*x*exp(-(x-0.4)*(x-0.4)/(2.0*1.00*1.00))
11.           + 0.25*x*x*exp(-(x+0.4)*(x+0.4)/(2.0*1.00*1.00));
12.   }
13.
14.   int main (int argc, char *argv[]) {
15.       int namelen, rank, size;
16.       char name[MPI_MAX_PROCESSOR_NAME];
17.       double upper_bound = 5.0, lower_bound = -5.0;
18.       double x0, x1, width;
19.       double integral = 0;
20.       double compute_time, total_time;
21.       int chunk_size;
22.       MPI_Init(&argc, &argv);
23.       MPI_Comm_size(MPI_COMM_WORLD, &size);
24.       MPI_Comm_rank(MPI_COMM_WORLD, &rank);
25.       MPI_Get_processor_name(name, &namelen);
26.       chunk_size = NUM_TRAPEZOIDS / size;
27.       x0 = lower_bound+(upper_bound-lower_bound)*rank/size;
28.       x1 = x0 + (upper_bound - lower_bound)/size;
29.       width = (x1-x0)/chunk_size;
30.       MPI_Barrier(MPI_COMM_WORLD);
31.       compute_time = total_time = MPI_Wtime();
32.   #pragma offload target(mic)
33.   #pragma omp parallel
34.   #pragma omp for reduction(+:integral)
35.       for (int i = 0; i < chunk_size ; i++) {
36.           integral += 0.5*width *(f(x0+width*i)+f(x0+width*(i+1)));
37.       }
38.       compute_time = MPI_Wtime() - compute_time;
39.       MPI_Allreduce(MPI_IN_PLACE, &integral, 1, MPI_DOUBLE, MPI_SUM,
      MPI_COMM_WORLD);
40.       total_time = MPI_Wtime() - total_time;
41.       printf("rank %d of %d on %s: %f seconds\n", rank, size, name,
      compute_time);
42.       if (rank == 0) {
43.           printf("integral = %f, time = %f\n", integral, total_time);
44.       }
45.       MPI_Finalize();
46.       return(0);
47.   }
```

HPMC 2022

■ Offloading to MIC

Both hpmc9 and hpmc7 are interconnected by infiniband.

```
#  13:17 root@HPMC9: /mm03fs/MPI # I_MPI_FABRICS=ofa mpirun -n 8 ./trap_offload

13:17 root@HPMC9: /mm03fs/MPI #  I_MPI_FABRICS=ofa mpirun -n 8 ./trap_offload
rank 0 of 8 on HPMC9: 10.728176 seconds
integral = 1.856399, time = 13.593669
rank 1 of 8 on HPMC9: 9.398290 seconds
rank 2 of 8 on HPMC9: 12.958419 seconds
rank 3 of 8 on HPMC9: 11.538105 seconds
rank 4 of 8 on HPMC9: 13.529295 seconds
rank 5 of 8 on HPMC9: 13.593640 seconds
rank 6 of 8 on HPMC9: 13.218497 seconds
rank 7 of 8 on HPMC9: 13.058236 seconds
```

■ Offloading to MIC with 2 Ranks

Running the MPI program, offloading by using the ofa (Open Fabrics Alliance).

```
#  13:36 root@HPMC9: /mm03fs/MPI # OMP_NUM_THREADS=8 I_MPI_FABRICS=ofa mpirun -n 2
```

./trap_offload

```
13:36 root@HPMC9: /mm03fs/MPI #  OMP_NUM_THREADS=8 I_MPI_FABRICS=ofa mpirun  -n 2 ./trap_offload
rank 0 of 2 on HPMC9: 22.342904 seconds
integral = 1.856399, time = 22.342928
rank 1 of 2 on HPMC9: 22.163031 seconds
```

Running the MPI progra, offloading by using dapl.

```
# 13:36 root@HPMC9: /mm03fs/MPI # OMP_NUM_THREADS=8
I_MPI_DAPL_PROVIDER_LIST=ofa-v2-mlx4_0-1u I_MPI_PERHOST=1 I_MPI_DEBUG=5
I_MPI_FABRICS=shm:dapl I_MPI_FALLBACK=0 /opt/INTEL-XE-2017-
update7/compilers_and_libraries_2017.7.259/linux/mpi/intel64/bin/mpiexec.hydra -
n 2 -hosts hpmc9 ./trap_offload
```

```
13:36 root@HPMC9: /mm03fs/MPI #  OMP_NUM_THREADS=8 I_MPI_DAPL_PROVIDER_LIST=ofa-v2-mlx4_0-1u I_MPI_PERHOST=1 I_MPI_DEBUG=5 I_MPI_FABR
[0] MPI startup(): Multi-threaded optimized library
[0] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[1] DAPL startup(): trying to open DAPL provider from I_MPI_DAPL_PROVIDER: ofa-v2-mlx4_0-1u
[0] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[0] MPI startup(): shm and dapl data transfer modes
[1] MPI startup(): DAPL provider ofa-v2-mlx4_0-1u
[1] MPI startup(): shm and dapl data transfer modes
[0] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[0] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[1] MPID_nem_init_dapl_coll_fns(): User set DAPL collective mask = 0000
[1] MPID_nem_init_dapl_coll_fns(): Effective DAPL collective mask = 0000
[0] MPI startup(): Rank    Pid      Node name  Pin cpu
[0] MPI startup(): 0       33344    HPMC9      {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,56,57,
                                    58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83}
[0] MPI startup(): 1       33345    HPMC9      {28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54
                                    ,55,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,
                                    107,108,109,110,111}
[0] MPI startup(): I_MPI_DAPL_PROVIDER_LIST=ofa-v2-mlx4_0-1u
[0] MPI startup(): I_MPI_DEBUG=5
[0] MPI startup(): I_MPI_FABRICS=shm:dapl
[0] MPI startup(): I_MPI_FALLBACK=0
[0] MPI startup(): I_MPI_INFO_NUMA_NODE_MAP=i40iw0:0,i40iw1:0,qib0:0,qib1:0,mic0:1
[0] MPI startup(): I_MPI_INFO_NUMA_NODE_NUM=2
[0] MPI startup(): I_MPI_PIN_MAPPING=2:0 0,1 28
rank 0 of 2 on HPMC9: 24.074761 seconds
integral = 1.856399, time = 24.215596
rank 1 of 2 on HPMC9: 24.215570 seconds
```

A Logging in to the MIC

MIC Essentials:

The MIC card is Intel Math Co-Processor that is slotted in PCIe and is made available on your HPMC host computer. The HPMC is the host computer and the MIC card is the client computer. At any time you should be able to ssh to the host MIC simply by typing: ssh mic0

This section introduces you to the essential administration of your MIC Co-processor-.

The MIC card is booted within your own HPMC as an embedded Linux system. There are many steps to get the MIC operational. Understanding these steps will help you to troubleshoot issues that you encounter when using your MIC Co-Processor.

First step, try to ssh to your MIC:

```
# ssh mic0
```

You should be logged into the MIC and presented with the MIC Linux prompt. Such login does not require to provide any user name or password. If you cannot login to your MIC, then make sure that MPSS is up and running:

```
# systemctl status mpss
```

You can restart MPSS and try to ssh mic0 again:

```
# systemctl restart mpss
```

Check that MPSS is up and serving, you can tail the dmesg in another terminal while "systemctl restart mpss" is being restarted:

```
# tail dmesg
```

If MPSS fails to restart then reset the MIC card:

```
# micctrl -rw
```

The above command will reset the MIC and wait until the reset is done. Now check on your MIC status. The MIC status must be in "ready" state before starting MPSS. Check the status of the MIC:

```
# micctrl -s
```

If the above command shows that the MIC is in "ready" state, then you can retry starting MPSS:

```
# systemctl restart mpss
```

Wait a minute then check if MPSS service is brought up successfully:

```
# systemctl status mpss
```

Also you can check the system dmesg output:

```
# dmesg
```

Now you should be able to ssh to your MIC:

```
# ssh mic0
```

If you cannot "ssh mic0" successfully, then you need to regenerate your id_rsa. Assuming you are the user 'root', issue the following command to regenerate your id_rsa:

```
# ssh-keygen
```

Add your /root/.ssh/id_rsa to /var/mpss/mic0/root/.ssh/authorized_keys:

```
# cat /root/.ssh/id_rsa >> /var/mpss/mic0/root/.ssh/authorized_keys
```

Restart MPSS:

```
# systemctl restart mpss
```

Wait until MPSS service is started successfully. Now you should be able to ssh to your MIC:

```
# ssh mic0
```

If you still have trouble accessing the MIC Co-Processor then you need to follow the step in the next appendix MIC TROUBLESHOOTING.

B Troubleshooting the MIC

The MIC is booted as a computer within your HPMC computer system. Your HPMC computer
is the host computer that will treat the MIC Co-Processor as a computer plugged into
its PCI bus, that will embed its OS image, for this to happen:
- the MIC device driver must be loaded, hence to communicate and control the MIC hardware
  layer. From your HPMC verify the mic module is loaded (this is true for x100 series)
  # lsmod | grep mic
- the MIC is recognized as a device to the host computer. You can simply see if the mic0
  device is there:
  # ls /dev/mic0
- the MPSS software layer must be available on the HPMC host computer. A simple way to
  verify that MPSS is installed, just see if micinfo can be recognized on your command
  prompt:
  # which micinfo

To bring the MIC up

* it is embedded system whose root directory is mapped to /var/mpss/mic0, hence the root
  user directory that will be available when MIC Co-Processor is bootstrapped can be
  seen on your HPMC host: /var/mpss/mic0/root
  # ls -la /var/mpss/mic0/root

Hence basic changes to the MIC root diretory are possible:
- Generate the key on the host and add it to the authorized_keys on the MIC mapped directory:
    ssh-keygen
    cat /root/.ssh/id_rsa.pub > /var/mpss/mic0/root/.ssh/authorized_keys
- micctrl --resetconfig -v
- check MIC if is in ready state
     micctrl -s
- if MIC is not in ready state then reset it
     micctrl -rw
- check MIC if is in ready state
     micctrl -s
- if MIC cannot be in ready state then you need to flash it (micflash -update) and reboot the host, and reset the MIC
  it is in ready state (mictrl -s)
- once the MIC is in ready state then bring up MPSS:
     systemctl start mpss
  you can monitor the progress of the startup: dmesg | tail
- once the MIC is booted, you can ssh to it:
     ssh mic0
- for physical condition and information about the MIC:
     micinfo
- enable MPSS service, so that the MPSS is up after each reboot. Assuming you won't make changes to the MIC kernel ima
     systemctl enable mpss

C Appendix Troubleshootig the CUDA Configuration

CUDA has been configured on the HPMC. The following environment variables are already exported by default.

```
export LPATH=$LPATH:/usr/lib/nvidia:/opt/bin:/opt/lib64:/opt/lib:/usr/local/cuda/lib64
export LIBRARY_PATH=$LIBRARY_PATH:/usr/lib/nvidia:/opt/lib64:/opt/lib:/usr/local/cuda/lib64
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib/nvidia:/opt/lib64:/opt/lib:/usr/local/cuda/lib64
export PATH=$PATH:/opt/bin:/opt/lib64:/opt/lib:/usr/local/cuda/bin:/usr/local/cuda/lib64
```

If you have issues running CUDA related applications, make sure that the following shell script is available /etc/ccube/st/cu9sysenv.sh

```
export LPATH=$LPATH:/usr/lib64/openmpi/lib:/usr/lib/nvidia:/opt/bin:/opt/lib64:/opt/lib:/usr/local/cuda-9.2/l
export LIBRARY_PATH=$LIBRARY_PATH:/usr/lib64/openmpi/lib:/usr/lib/nvidia:/opt/lib64:/opt/lib:/usr/local/cuda-

export LUA_PATH='/cuda9/.luarocks/share/lua/5.1/?.lua;
/cuda9/.luarocks/share/lua/5.1/?/init.lua;
/cuda9/torch/install/share/lua/5.1/?.lua;
/cuda9/torch/install/share/lua/5.1/?/init.lua;
./?.lua;/cuda9/torch/install/share/luajit-2.1.0-beta1/?.lua;
/usr/local/share/lua/5.1/?.lua;/usr/local/share/lua/5.1/?/init.lua'

export LUA_CPATH='/cuda9/.luarocks/lib/lua/5.1/?.so;
/cuda9/torch/install/lib/lua/5.1/?.so;
/cuda9/torch/install/lib/?.so;./?.so;
/usr/local/lib/lua/5.1/?.so;/usr/local/lib/lua/5.1/loadall.so'

export PATH=$PATH:/usr/local/cuda-9.2/bin:/usr/local/cuda-9.2/lib64:/cuda9/CuViews:
/cuda9/CuExamples:/cuda9/CuExamples/comp:/cuda9/torch/install/bin:
/tools:/usr/local/ccube/bin:/usr/local/pgplot:$JPATH/bin

export LD_LIBRARY_PATH="${LD_LIBRARY_PATH:+$LD_LIBRARY_PATH:}/cuda9/torch/install/lib:
/usr/local/lib64:/usr/lib64/openmpi/lib:/usr/local/cuda-9.2/lib64:
/usr/local/cuda-9.2/libnvvp:/usr/local/cuda-9.2/libnsight:$JPATH/bin:$JPATH/jre/bin/classic"

export DYLD_LIBRARY_PATH=/cuda9/torch/install/lib

export LUA_CPATH='/cuda9/torch/install/lib/?.so;'$LUA_CPATH

# test CUDNN with layer.lua and bench.lua
export CUDNN_PATH=/usr/local/cuda-9.2/lib64/libcudnn.so.7.1.4

# NOTE the following can be overriden in user .bashrc or .profile in case using /mm03fs/miclib17, also source
#    but also need to change sym link of /opt/intel and may change /var/mpss/mic0/~/.bashrc
#export SINK_LD_LIBRARY_PATH=/mm03fs/miclib13
export SINK_LD_LIBRARY_PATH=/mc00/miclib13
```

D A Simple OMP Program

Consider the following simple program to measure the DP GFLops by running many threads over OMP.

-- Program Code D.1 : [LISTING gigaflop.cpp] - [A simple OMP program gigaflop.cpp]

(raw text)

```
1.     #include <stdio.h>
2.     #include <stdlib.h>
3.     #include <omp.h>
4.
5.     unsigned int const SIZE=16;
6.     unsigned int const ITER=48000000;
7.
8.     extern double elapsedTime(void);
9.
10.    int main()
11.    {
12.        double startTime, duration;
13.        int i;
14.        __declspec(aligned(64)) double a[SIZE],b[SIZE],c[SIZE];
15.        // initialize
16.        for (i=0;i<SIZE;i++) {
17.            c[i]=b[i]=a[i]=(double)rand();
18.        }
19.        // warmup
20.        #pragma omp parallel for
21.        for(i=0;i<ITER;i++) {
22.            #pragma vector aligned(a,b,c)
23.            a[0:SIZE]=b[0:SIZE]*c[0:SIZE]+a[0:SIZE];
24.        }
25.        startTime = elapsedTime();
26.        #pragma omp parallel for
27.        for(i=0;i<ITER;i++) {
28.            #pragma vector aligned(a,b,c)
29.            a[0:SIZE]=b[0:SIZE]*c[0:SIZE]+a[0:SIZE];
30.        }
31.        duration = elapsedTime() - startTime;
32.        double Gflop = 2*SIZE*ITER/1e+9;
33.        double Gflops = Gflop/duration;
34.        printf("Running %d openmp threads\n", omp_get_max_threads());
35.        printf("DP GFlops = %f\n", Gflops);
36.        return 0;
37.    }
```

HPMC 2022

Running the program on the HPMC's MIC coprocessor:

12:46 root@HPMC9: ~/coding # xphi gigaflop_xphi

```
12:46 root@HPMC9: ~/coding #  xphi gigaflop_xphi
Running 224 openmp threads
DP GFlops = 154310202251.497009
```

Running the program on the HPMC host:

12:50 root@HPMC9: ~/coding # ./gigaflop_hos

```
12:50 root@HPMC9: ~/coding #  ./gigaflop_hos
Running 112 openmp threads
DP GFlops = 139649512.149654
```

E dint: display the system interrupts

Your HPMC comes with the dint program that display the system interrupts. dint is a command that you can issue at the command prompt. Since your HPMC has hundreds of cores, and your Xeon Phi MIC has as well hundreds of cores, you need a utility that can display the system interrupts while focusing on the busiest cores.

The chapter "Programming and Running by MIC Native Loading" showed how to run diffusion_omp by offloading it to the MIC. In the following we show how to use the dint command to display the interrupts as the program is executing.

The interrupt names may be different for each system. To display the names of the interrupts, you can simply use dint with the -info option:

```
# dint -info
```

dint can display all cores. Use -transpose option so that the 228 cores are displayed one per row, a neater view.

```
# (HPMC9) 03:59 root@mic0: /tools # ./dint -transpose -iter 7 -delay 1 -interrupts
LOC,RES,CAL,4,5,26,TLB
```

Use the -prune option to display the cores by pruning a range between the beginning and the end:

```
# (HPMC9) 03:59 root@mic0: /tools # ./dint -transpose -iter 7 -delay 1 -interrupts
LOC,RES,CAL,4,5,26,TLB -prune 3,5
```

Displaying all the cores can be distracting, so to focus only on a range of cores, just add in -sortby option:

```
# (HPMC9) 04:02 root@mic0: /tools # ./dint -transpose -iter 7 -delay 1 -interrupts
LOC,RES,CAL,4,5,26,TLB -prune 3,5 -sortby RES
```

To print all cores focusing on the RES interrupt, use this command:

```
# (HPMC9) 03:59 root@mic0: /tools # ./dint -transpose -iter 7 -delay 1 -interrupts
LOC,RES,CAL,4,5,26,TLB -sortby RES
```

Use the -transpose option so that each core or numbered-CPU is displayed on a row.

The following shows the dint command running on the MIC of hpmc9, where we focused on five cores that are busy with the RES (Rescheduling interrupts). After each iteration the busiest cores with the RES are displayed first.

```
# (HPMC9) 03:40 root@mic0: /tools # ./dint -NVU 5 -iter 7 -delay 1 -interrupts LOC,RES,CAL,4,5,26,TLB
-sortby RES

(HPMC9) 03:40 root@mic0: /tools #  ./dint -NVU 5 -iter 7 -delay 1 -interrupts LOC,RES,CAL,4,5,26,TLB -sortby R
                           CPU227      CPU0    CPU225    CPU167     CPU26    summed
2026-04-23 03:41:04     LOC        70       121        80       162       162       595     Local timer interr
2026-04-23 03:41:04     RES        12         8         3         2         2        27     Rescheduling inter
2026-04-23 03:41:04     CAL        84         0        47         0         0       131     Function call inte
2026-04-23 03:41:04       4         0         0         0         0         0         0     IO-APIC-edge
2026-04-23 03:41:04       5         0         0         0         0         0         0     IO-APIC-edge
2026-04-23 03:41:04      26         0         0         0         0         0         0     SBOX-ICR-edge
2026-04-23 03:41:04     TLB         0         0         1         0         0         1     TLB shootdowns

                           CPU226    CPU227      CPU0    CPU225    CPU119    summed
2026-04-23 03:41:06     LOC       118       132       150        62       159       621     Local timer interr
2026-04-23 03:41:06     RES        12         9         8         7         3        39     Rescheduling inter
2026-04-23 03:41:06     CAL         0         4         8        93         0       105     Function call inte
2026-04-23 03:41:06       4         0         0         2         0         0         2     IO-APIC-edge
2026-04-23 03:41:06       5         0         0         0         0         0         0     IO-APIC-edge
2026-04-23 03:41:06      26         0         0         4         0         0         4     SBOX-ICR-edge
2026-04-23 03:41:06     TLB         0         0         0         1         0         1     TLB shootdowns

                           CPU227      CPU0    CPU225    CPU226     CPU10    summed
```

```
2026-04-23 03:41:07     LOC        96       101        40        71       156       464     Local timer interr
2026-04-23 03:41:07     RES        10         9         5         4         2        30     Rescheduling inter
2026-04-23 03:41:07     CAL        33        53        47         0         0       133     Function call inte
2026-04-23 03:41:07       4         0         2         0         0         0         2     IO-APIC-edge
2026-04-23 03:41:07       5         0         0         0         0         0         0     IO-APIC-edge
2026-04-23 03:41:07      26         0         3         0         0         0         3     SBOX-ICR-edge
2026-04-23 03:41:07     TLB         0         0         1         2         0         3     TLB shootdowns

                                 CPU0    CPU225    CPU227    CPU226    CPU134    summed
2026-04-23 03:41:09     LOC        99       149        98       101       160       607     Local timer interr
2026-04-23 03:41:09     RES        18        13        11         5         3        50     Rescheduling inter
2026-04-23 03:41:09     CAL        49        11         0        49         0       109     Function call inte
2026-04-23 03:41:09       4         2         0         0         0         0         2     IO-APIC-edge
2026-04-23 03:41:09       5         0         0         0         0         0         0     IO-APIC-edge
2026-04-23 03:41:09      26         4         0         0         0         0         4     SBOX-ICR-edge
2026-04-23 03:41:09     TLB         1         0         1         1         0         3     TLB shootdowns

                               CPU226    CPU227      CPU0    CPU225    CPU118    summed
2026-04-23 03:41:10     LOC        58        66       138       146       157       565     Local timer interr
2026-04-23 03:41:10     RES        10         8         7         4         3        32     Rescheduling inter
2026-04-23 03:41:10     CAL       103        35         0         0         0       138     Function call inte
2026-04-23 03:41:10       4         0         0         2         0         0         2     IO-APIC-edge
2026-04-23 03:41:10       5         0         0         0         0         0         0     IO-APIC-edge
2026-04-23 03:41:10      26         0         0         4         0         0         4     SBOX-ICR-edge
2026-04-23 03:41:10     TLB         0         0         0         1         0         1     TLB shootdowns

                               CPU226    CPU227      CPU0    CPU225     CPU34    summed
2026-04-23 03:41:12     LOC       130        81       106       105       160       582     Local timer interr
2026-04-23 03:41:12     RES        15        10         8         3         2        38     Rescheduling inter
2026-04-23 03:41:12     CAL         0        58         0        22         0        80     Function call inte
2026-04-23 03:41:12       4         0         0         2         0         0         2     IO-APIC-edge
2026-04-23 03:41:12       5         0         0         0         0         0         0     IO-APIC-edge
2026-04-23 03:41:12      26         0         0         4         0         0         4     SBOX-ICR-edge
2026-04-23 03:41:12     TLB         0         0         0         1         0         1     TLB shootdowns

                               CPU226    CPU227      CPU0    CPU225      CPU4    summed
2026-04-23 03:41:13     LOC       139        76        91        88       158       552     Local timer interr
2026-04-23 03:41:13     RES        14         9         8         6         2        39     Rescheduling inter
2026-04-23 03:41:13     CAL         0        50         3        74         0       127     Function call inte
2026-04-23 03:41:13       4         0         0         2         0         0         2     IO-APIC-edge
2026-04-23 03:41:13       5         0         0         0         0         0         0     IO-APIC-edge
2026-04-23 03:41:13      26         0         0         4         0         0         4     SBOX-ICR-edge
2026-04-23 03:41:13     TLB         0         0         1         2         0         3     TLB shootdowns
```

While the above command showed the interrupts activity on the MIC, the following command shows the interrupt activity on the host hpmc9. It is captured while an OMP program was running over the xphi.

```
# 03:43 root@HPMC9: /tools # ./dint -NVU 5 -iter 7 -delay 1 -interrupts LOC,RES,CAL,TLB -sortby RES

03:43 root@HPMC9: /tools #  ./dint -NVU 5 -iter 7 -delay 1 -interrupts LOC,RES,CAL,TLB -sortby RES
                                 CPU0      CPU5     CPU43     CPU37     CPU34    summed
2026-04-23 03:44:00     LOC         8         2         0         1         4        15     Local timer interr
2026-04-23 03:44:00     RES         6         0         0         0         0         6     Rescheduling inter
2026-04-23 03:44:00     CAL         0         0         0         0         0         0     Function call inte
2026-04-23 03:44:00     TLB         0         0         0         0         0         0     TLB shootdowns

                                 CPU0      CPU5     CPU43     CPU37     CPU34    summed
2026-04-23 03:44:01     LOC         4         2         3         3         0        12     Local timer interr
2026-04-23 03:44:01     RES         3         0         0         0         0         3     Rescheduling inter
2026-04-23 03:44:01     CAL         0         0         0         0         0         0     Function call inte
2026-04-23 03:44:01     TLB         0         0         0         0         0         0     TLB shootdowns

                                 CPU0     CPU30      CPU5     CPU43     CPU37    summed
2026-04-23 03:44:02     LOC         7        56         4         0         3        70     Local timer interr
2026-04-23 03:44:02     RES         2         1         0         0         0         3     Rescheduling inter
2026-04-23 03:44:02     CAL         0         0         0         0         0         0     Function call inte
2026-04-23 03:44:02     TLB         0         0         0         0         0         0     TLB shootdowns

                                 CPU0      CPU5     CPU43     CPU37     CPU34    summed
2026-04-23 03:44:03     LOC         5         2         2         2         3        14     Local timer interr
2026-04-23 03:44:03     RES         2         0         0         0         0         2     Rescheduling inter
2026-04-23 03:44:03     CAL         0         0         0         0         0         0     Function call inte
2026-04-23 03:44:03     TLB         0         0         0         0         0         0     TLB shootdowns

                                 CPU0      CPU5     CPU43     CPU37     CPU34    summed
2026-04-23 03:44:04     LOC         4         4         3         3         5        19     Local timer interr
2026-04-23 03:44:04     RES         5         0         0         0         0         5     Rescheduling inter
2026-04-23 03:44:04     CAL         0         0         0         0         0         0     Function call inte
2026-04-23 03:44:04     TLB         0         0         0         0         0         0     TLB shootdowns

                                 CPU0     CPU30      CPU5     CPU43     CPU37    summed
2026-04-23 03:44:05     LOC        10        52         5         0         0        67     Local timer interr
2026-04-23 03:44:05     RES         2         2         0         0         0         4     Rescheduling inter
2026-04-23 03:44:05     CAL         0         0         0         0         0         0     Function call inte
2026-04-23 03:44:05     TLB         0         0         0         0         0         0     TLB shootdowns

                                 CPU0      CPU5     CPU43     CPU37     CPU34    summed
2026-04-23 03:44:06     LOC         6         4         2         2         6        20     Local timer interr
2026-04-23 03:44:06     RES         3         0         0         0         0         3     Rescheduling inter
```

2026-04-23 03:44:06	CAL	0	0	0	0	0	0	Function call inte
2026-04-23 03:44:06	TLB	0	0	0	0	0	0	TLB shootdowns

F dcpus: display the system stats

Your HPMC comes with the dcpus program that display the system stats. dcpus is a command that you issue at the command prompt. Since your HPMC has hundreds of cores, and your Xeon Phi MIC has as well hundreds of cores, you need a utility that can display the system stats while focusing on the busiest cores.

The chapter "Programming and Running by MIC Native Loading" showed how to run diffusion_omp by offloading it to the MIC. In the following we show how to use the dcpus command to display the stat of the cores as the program is executing.

To display the stats names on your system, you can simply use dcpus with the -info option:

```
# dcpus -info
```

dcpus can display all cores. Use -transpose option so that the 228 cores are displayed one per row, a neater view.

```
# (HPMC9) 03:59 root@mic0: /tools # ./dcpus -transpose -iter 7 -delay 1
```

Use the -prune option to display the cores by pruning a range between the beginning and the end:

```
# (HPMC9) 03:59 root@mic0: /tools # ./dcpus -transpose -iter 7 -delay 1 -prune 3,5
```

Displaying all the cores can be distracting, so to focus only on a range of cores, just add in -sortby option:

```
# (HPMC9) 04:02 root@mic0: /tools # ./dcpus -transpose -iter 7 -delay 1 -prune 3,5 -sortby system
```

Use the -transpose option so that each core or numbered-CPU is displayed on a row.

To print all cores focusing on the RES interrupt, use this command:

```
# (HPMC9) 03:59 root@mic0: /tools # ./dcpus -transpose -iter 7 -delay 1 -sortby system
```

The following shows the dcpus command running on the MIC of hpmc9, where we focused on eight cores that are busy with the RES (Rescheduling interrupts). After each iteration the busiest cores with the system stat are displayed first. We used -prune to select the only the first three cores and the last five cores. The dcpus was started few seconds before launching the "xphi diffusion_omp_xphi", so the usage of the MIC went from 0 to 98%, showing all core stat activities.

```
# (HPMC9) 13:56 root@mic0: /tools # ./dcpus -transpose -iter 7 -delay 1 -prune 3,5 -sortby idle
```

```
(HPMC9) 13:56 root@mic0: /tools #  ./dcpus -transpose -iter 7 -delay 1 -prune 3,5 -sortby idle
                                    user      nice    system      idle    iowait       irq   softirq     steal
2026-04-23 14:14:34      CPU30         0         0         0       218         0         0         0         0
2026-04-23 14:14:34      CPU29         0         0         0       213         0         0         1         0
2026-04-23 14:14:34      CPU26         0         0         0       207         0         0         0         0
                           ...
2026-04-23 14:14:34      CPU13         0         0         0        86         0         0         0         0
2026-04-23 14:14:34      CPU16         0         0         0        78         0         0         0         0
2026-04-23 14:14:34      CPU18         0         0         0        72         0         0         0         0
2026-04-23 14:14:34      CPU22         0         0        40        31         0         0         0         0
2026-04-23 14:14:34      CPU14         0         0         0        11         0         0         0         0
                        summed         3         0        42     32427         0         0        18         0
used=63  idle=32427  usage 0.193905817174515%
                                    user      nice    system      idle    iowait       irq   softirq     steal
2026-04-23 14:14:35      CPU14         0         0         0       259         0         0         0         0
2026-04-23 14:14:35     CPU109         0         0         0       205         0         0         0         0
2026-04-23 14:14:35     CPU213         0         0         0       204         0         0         0         0
                           ...
2026-04-23 14:14:35     CPU165         0         0         0       101         0         0         1         0
2026-04-23 14:14:35     CPU125         0         0         0       101         0         0         1         0
2026-04-23 14:14:35      CPU28         0         0         0        94         0         0         0         0
2026-04-23 14:14:35      CPU26         0         0         0        94         0         0         0         0
2026-04-23 14:14:35      CPU30         0         0        40        51         0         0         0         0
                        summed         7         0        41     34300         0         0        21         0
used=69  idle=34300  usage 0.200762314876779%
                                    user      nice    system      idle    iowait       irq   softirq     steal
2026-04-23 14:14:37     CPU169         0         0         0       204         0         0         0         0
2026-04-23 14:14:37      CPU36         0         0         0       204         0         0         0         0
2026-04-23 14:14:37      CPU40         0         0         0       204         0         0         0         0
                           ...
2026-04-23 14:14:37      CPU18         0         0         0       102         0         0         0         0
2026-04-23 14:14:37     CPU118         0         0         0       102         0         0         0         0
2026-04-23 14:14:37      CPU90         0         0         0       102         0         0         1         0
```

```
2026-04-23 14:14:37     CPU181         0         0         0       101         0         0         4         0
2026-04-23 14:14:37     CPU220         0         0         0       101         0         0         0         0
                        summed         7         0        39     32906         0         0        15         0
used=61  idle=32906  usage 0.185033518366852%
                                    user      nice    system      idle    iowait       irq   softirq     steal
2026-04-23 14:14:38      CPU26        32         0         2       214         0         0         0         0
2026-04-23 14:14:38     CPU158        33         0         1       209         0         0         0         0
2026-04-23 14:14:38      CPU91        33         0         1       209         0         0         0         0
                           ...
2026-04-23 14:14:38      CPU38        33         0         1       114         0         0         0         0
2026-04-23 14:14:38      CPU42        33         0         1       114         0         0         0         0
2026-04-23 14:14:38     CPU193        32         0         2       114         0         0         0         0
2026-04-23 14:14:38      CPU45        39         0         2       107         0         0         0         0
2026-04-23 14:14:38      CPU46        33         0        40        75         0         0         0         0
                        summed      7368         0       386     36271         0         0        14         0
used=7768  idle=36271  usage 17.6389109652808%
                                    user      nice    system      idle    iowait       irq   softirq     steal
2026-04-23 14:14:40     CPU225         0         0         0       156         0         0         0         0
2026-04-23 14:14:40       CPU0         9         0         0       146         0         0         0         0
2026-04-23 14:14:40     CPU226         0         0        46       110         0         0         0         0
                           ...
2026-04-23 14:14:40     CPU206       149         0         3         4         0         0         0         0
2026-04-23 14:14:40      CPU90       151         0         1         4         0         0         0         0
2026-04-23 14:14:40      CPU32       149         0         2         4         0         0         0         0
2026-04-23 14:14:40     CPU127       150         0         2         4         0         0         0         0
2026-04-23 14:14:40       CPU1       151         0         4         0         0         0         0         0
                        summed     33576         0       543      1406         0         0         0         0
used=34119  idle=1406  usage 96.0422237860661%
                                    user      nice    system      idle    iowait       irq   softirq     steal
2026-04-23 14:14:42     CPU225         1         0         0       153         0         0         0         0
2026-04-23 14:14:42       CPU0         8         0         0       146         0         0         0         0
2026-04-23 14:14:42     CPU226         0         0        44       109         0         0         0         0
                           ...
2026-04-23 14:14:42     CPU118       152         0         1         0         0         0         0         0
2026-04-23 14:14:42     CPU206       152         0         1         0         0         0         0         0
2026-04-23 14:14:42      CPU90       152         0         1         0         0         0         0         0
2026-04-23 14:14:42      CPU32       154         0         0         0         0         0         0         0
2026-04-23 14:14:42     CPU127       153         0         0         0         0         0         0         0
                        summed     34163         0       280       468         0         0         0         0
used=34443  idle=468  usage 98.6594483114205%
                                    user      nice    system      idle    iowait       irq   softirq     steal
2026-04-23 14:14:43     CPU227         0         0        32       160         0         0         0         0
2026-04-23 14:14:43       CPU0         6         0         0       149         0         0         0         0
2026-04-23 14:14:43     CPU225         0         0        45       111         0         0         0         0
                           ...
2026-04-23 14:14:43     CPU118       156         0         0         0         0         0         0         0
2026-04-23 14:14:43     CPU206       156         0         0         0         0         0         0         0
2026-04-23 14:14:43      CPU90       154         0         2         0         0         0         0         0
2026-04-23 14:14:43      CPU32       154         0         1         0         0         0         0         0
2026-04-23 14:14:43     CPU127       154         0         2         0         0         0         0         0
                        summed     34726         0       293       528         0         0         0         0
used=35019  idle=528  usage 98.5146425858722%
```

While the above command showed the interrupts activity on the MIC, the following command shows the stats on the host hpmc9. It is captured while an OMP program was running over the xphi. Notice that the system is idle since the xphi dispatched all the work to the MIC.

```
# 14:14 root@HPMC9: /tools # ./dcpus -transpose -iter 7 -delay 1 -prune 3,5

14:14 root@HPMC9: /tools #  ./dcpus -transpose -iter 7 -delay 1 -prune 3,5
                                    user      nice    system      idle    iowait       irq   softirq     steal
2026-04-23 14:15:21       CPU0         0         0         0       101         0         0         0         0
2026-04-23 14:15:21       CPU1         0         0         0       101         0         0         0         0
2026-04-23 14:15:21       CPU2         1         0         0       100         0         0         0         0
                           ...
2026-04-23 14:15:21     CPU107         0         0         0       101         0         0         0         0
2026-04-23 14:15:21     CPU108         0         0         0       101         0         0         0         0
2026-04-23 14:15:21     CPU109         0         0         0       101         0         0         0         0
2026-04-23 14:15:21     CPU110         0         0         0       100         0         0         0         0
2026-04-23 14:15:21     CPU111         0         0         0       101         0         0         0         0
                        summed         1         0         0     11284         0         0         0         0
used=1  idle=11284  usage 0.00886132033673017%
                                    user      nice    system      idle    iowait       irq   softirq     steal
2026-04-23 14:15:22       CPU0         0         0         0       101         0         0         0         0
2026-04-23 14:15:22       CPU1         0         0         0       101         0         0         0         0
2026-04-23 14:15:22       CPU2         0         0         0       100         0         0         0         0
                           ...
2026-04-23 14:15:22     CPU107         0         0         0       101         0         0         0         0
2026-04-23 14:15:22     CPU108         0         0         0       101         0         0         0         0
2026-04-23 14:15:22     CPU109         0         0         0       101         0         0         0         0
2026-04-23 14:15:22     CPU110         0         0         0       102         0         0         0         0
2026-04-23 14:15:22     CPU111         0         0         0       101         0         0         0         0
                        summed         0         0         1     11331         0         0         0         0
used=1  idle=11331  usage 0.00882456759618779%
                                    user      nice    system      idle    iowait       irq   softirq     steal
2026-04-23 14:15:23       CPU0         0         0         0       101         0         0         0         0
2026-04-23 14:15:23       CPU1         0         0         0       102         0         0         0         0
2026-04-23 14:15:23       CPU2         1         0         1       101         0         0         0         0
                           ...
2026-04-23 14:15:23     CPU107         0         0         0       101         0         0         0         0
2026-04-23 14:15:23     CPU108         0         0         0       101         0         0         0         0
2026-04-23 14:15:23     CPU109         0         0         0       101         0         0         0         0
2026-04-23 14:15:23     CPU110         0         0         0       101         0         0         0         0
2026-04-23 14:15:23     CPU111         0         0         0       101         0         0         0         0
                        summed         1         0         1     11339         0         0         0         0
used=2  idle=11339  usage 0.0176351291773212%
                                    user      nice    system      idle    iowait       irq   softirq     steal
```

```
2026-04-23 14:15:24       CPU0         0         0         0       101         0         0         0         0
2026-04-23 14:15:24       CPU1         0         0         0       101         0         0         0         0
2026-04-23 14:15:24       CPU2         1         0         0       100         0         0         0         0
                           ...
2026-04-23 14:15:24     CPU107         0         0         0       102         0         0         0         0
2026-04-23 14:15:24     CPU108         0         0         0       102         0         0         0         0
2026-04-23 14:15:24     CPU109         0         0         0       101         0         0         0         0
2026-04-23 14:15:24     CPU110         0         0         0       101         0         0         0         0
2026-04-23 14:15:24     CPU111         0         0         0       101         0         0         0         0
                        summed         1         0         2     11323         0         0         0         0
used=3  idle=11323  usage 0.0264877273529931%
                                    user      nice    system      idle    iowait       irq   softirq     steal
2026-04-23 14:15:25       CPU0         0         0         0       101         0         0         0         0
2026-04-23 14:15:25       CPU1         0         0         0       101         0         0         0         0
2026-04-23 14:15:25       CPU2         0         0         0       101         0         0         0         0
                           ...
2026-04-23 14:15:25     CPU107         0         0         0       101         0         0         0         0
2026-04-23 14:15:25     CPU108         0         0         0       101         0         0         0         0
2026-04-23 14:15:25     CPU109         0         0         0       102         0         0         0         0
2026-04-23 14:15:25     CPU110         0         0         0       101         0         0         0         0
2026-04-23 14:15:25     CPU111         0         0         0       102         0         0         0         0
                        summed         1         0         0     11331         0         0         0         0
used=1  idle=11331  usage 0.00882456759618779%
                                    user      nice    system      idle    iowait       irq   softirq     steal
2026-04-23 14:15:26       CPU0         0         0         0       101         0         0         0         0
2026-04-23 14:15:26       CPU1         0         0         0       101         0         0         0         0
2026-04-23 14:15:26       CPU2         1         0         0       100         0         0         0         0
                           ...
2026-04-23 14:15:26     CPU107         0         0         0       101         0         0         0         0
2026-04-23 14:15:26     CPU108         0         0         0       101         0         0         0         0
2026-04-23 14:15:26     CPU109         0         0         0       101         0         0         0         0
2026-04-23 14:15:26     CPU110         0         0         0       101         0         0         0         0
2026-04-23 14:15:26     CPU111         0         0         0       101         0         0         0         0
                        summed         1         0         1     11332         0         0         0         0
used=2  idle=11332  usage 0.0176460208223046%
                                    user      nice    system      idle    iowait       irq   softirq     steal
2026-04-23 14:15:27       CPU0         0         0         0       101         0         0         0         0
2026-04-23 14:15:27       CPU1         0         0         0       101         0         0         0         0
2026-04-23 14:15:27       CPU2         1         0         0       100         0         0         0         0
                           ...
2026-04-23 14:15:27     CPU107         0         0         0       101         0         0         0         0
2026-04-23 14:15:27     CPU108         0         0         0       101         0         0         0         0
2026-04-23 14:15:27     CPU109         0         0         0       101         0         0         0         0
2026-04-23 14:15:27     CPU110         0         0         0       101         0         0         0         0
2026-04-23 14:15:27     CPU111         0         0         0       101         0         0         0         0
                        summed         1         0         0     11327         0         0         0         0
used=1  idle=11327  usage 0.00882768361581921%
```

G How to display the system interrupts of your HPMC

Use the dint command to display the interrupts of your HPMC computer.

17:31 root@HPMC7: ~ # dint -info

```
17:31 root@HPMC7: ~ #  dint -info
  CPU COUNT 272

     0    IR-IO-APIC-edge      timer
     8    IR-IO-APIC-edge      rtc0
     9    IR-IO-APIC-fasteoi   acpi
    18    IR-IO-APIC-fasteoi   ehci_hcd:usb1, ehci_hcd:usb2, i801_smbus
    25    IR-PCI-MSI-edge      PCIe PME
    27    IR-PCI-MSI-edge      PCIe PME
    29    IR-PCI-MSI-edge      PCIe PME
    30    IR-PCI-MSI-edge      PCIe PME
    31    IR-PCI-MSI-edge      PCIe PME
    32    IR-PCI-MSI-edge      xhci_hcd
    33    IR-PCI-MSI-edge      0000:00:11.4
    34    IR-PCI-MSI-edge      mic
    35    IR-PCI-MSI-edge      ib_qib0
    36    IR-PCI-MSI-edge      ib_qib0 (buf avail)
    37    IR-PCI-MSI-edge      ib_qib0 (sdma 0)
    38    IR-PCI-MSI-edge      ib_qib0 (sdmaI 0)
    39    IR-PCI-MSI-edge      ib_qib0 (sdmaP 0)
    40    IR-PCI-MSI-edge      ib_qib0 (sdmaC 0)
    41    IR-PCI-MSI-edge      ib_qib0 (kctx)
    42    IR-IO-APIC-fasteoi   snd_hda_intel
    43    IR-PCI-MSI-edge      enp5s0f0
    44    IR-PCI-MSI-edge      enp5s0f0-TxRx-0
    45    IR-PCI-MSI-edge      enp5s0f0-TxRx-1
    46    IR-PCI-MSI-edge      enp5s0f0-TxRx-2
    47    IR-PCI-MSI-edge      enp5s0f0-TxRx-3
    48    DMAR_MSI-edge      dmar0
    49    IR-PCI-MSI-edge      enp5s0f0-TxRx-4
    50    IR-PCI-MSI-edge      enp5s0f0-TxRx-5
    51    IR-PCI-MSI-edge      enp5s0f0-TxRx-6
    52    IR-PCI-MSI-edge      enp5s0f0-TxRx-7
    53    IR-PCI-MSI-edge      enp5s0f1
    54    IR-PCI-MSI-edge      enp5s0f1-TxRx-0
    55    IR-PCI-MSI-edge      enp5s0f1-TxRx-1
    56    IR-PCI-MSI-edge      enp5s0f1-TxRx-2
    57    IR-PCI-MSI-edge      enp5s0f1-TxRx-3
    58    IR-PCI-MSI-edge      enp5s0f1-TxRx-4
    59    IR-PCI-MSI-edge      enp5s0f1-TxRx-5
    60    IR-PCI-MSI-edge      enp5s0f1-TxRx-6
    61    IR-PCI-MSI-edge      enp5s0f1-TxRx-7
  NMI    Non-maskable interrupts
  LOC    Local timer interrupts
  SPU    Spurious interrupts
  PMI    Performance monitoring interrupts
  IWI    IRQ work interrupts
  RTR    APIC ICR read retries
  RES    Rescheduling interrupts
  CAL    Function call interrupts
  TLB    TLB shootdowns
  TRM    Thermal event interrupts
  THR    Threshold APIC interrupts
  DFR    Deferred Error APIC interrupts
  MCE    Machine check exceptions
  MCP    Machine check polls
  PIN    Posted-interrupt notification event
  PIW    Posted-interrupt wakeup event
```

H How to display the interrupts of the MIC

To display the interrupts of the Xeon Phi MIC, ssh to the MIC and issue the dint command.

 # (HPMC7) 17:50 root@mic0: /tools # ./dint -info

```
(HPMC7) 17:50 root@mic0: /tools #  ./dint -info
  CPU COUNT 228

  INTERRUPTS MNEMONICS: 4 5 6 7 13 17 26 27 28 30 NMI LOC SPU PMI IWI RES CAL TLB TRM THR MCE MCP

     4    IO-APIC-edge      dma channel
     5    IO-APIC-edge      dma channel
     6    IO-APIC-edge      dma channel
     7    IO-APIC-edge      dma channel
    13    IO-APIC-edge      TMU
    17    IO-APIC-edge      SCIF INTR 0
    26    SBOX-ICR-edge      vnet intr
    27    SBOX-ICR-edge      Shutdown
    28    SBOX-ICR-edge      hvc intr
    30    SBOX-ICR-edge      PC35_Wakeup
   NMI    Non-maskable interrupts
   LOC    Local timer interrupts
   SPU    Spurious interrupts
   PMI    Performance monitoring interrupts
   IWI    IRQ work interrupts
   RES    Rescheduling interrupts
   CAL    Function call interrupts
   TLB    TLB shootdowns
   TRM    Thermal event interrupts
   THR    Threshold APIC interrupts
   MCE    Machine check exceptions
   MCP    Machine check polls
```