

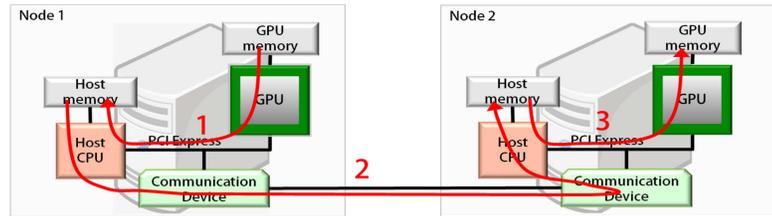
# An MPI Library implementing Direct Communication for Many-Core based Accelerators

Min Si and Yutaka Ishikawa (Advisor), University of Tokyo, Tokyo, Japan

## DCFA (Direct Communication Facility Architecture)

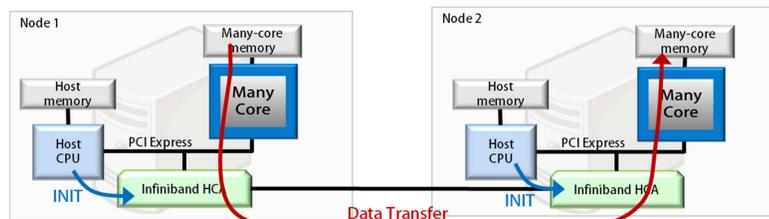
### Communication Issues on GPU

- An accelerator is a PCI-Express device, and thus it cannot configure/initialize another device such as a communication device.
- Though the PCI-Express address is known by a GPU, the GPU cannot issue commands to a communication device.
- The current Mellanox GPU Direct technology does not provide direct communication between GPUs, but **data is copied to the memory in the Host CPU and then transferred to the remote Host.**  
( GPU Direct v2.0 provides direct communication between the GPUs in one node )



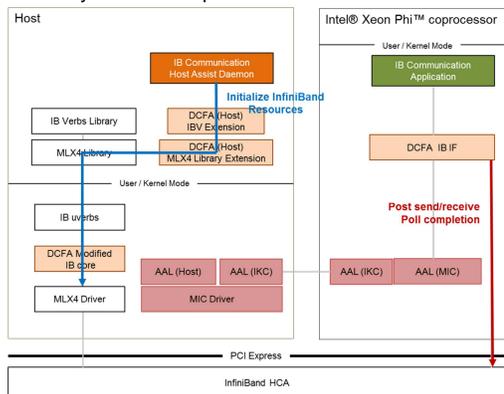
### Communication on Intel® Xeon Phi™ coprocessors ( DCFA )

- Host CPU initializes Infiniband HCA, and relays the PCI-Express address to Intel® Xeon Phi™ coprocessors .
- Internal structures of the Infiniband HCA are distributed to both the memory space of the host and that of the Intel® Xeon Phi™ coprocessors.
- Intel® Xeon Phi™ coprocessors communicate directly by issuing commands to the Infiniband HCA.**



### Design Details

- Based on the AAL (Accelerator Abstraction Layer)
  - Hides hardware-specific functions and provides kernel programming interfaces to operating system developers



- ✓ AAL(Host)  
A Linux device driver provides operation functions for the host
- ✓ AAL(Many-core) :  
Provides operation functions for the micro kernel on Intel® Xeon Phi™ coprocessor.
- ✓ AAL(IKC)  
Provides communication between a Host and an Intel® Xeon Phi™ coprocessor.

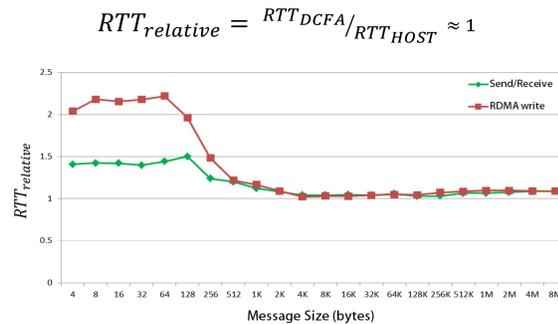
- DCFA (Host) IBV / MLX4 Library Extension  
Provides IB initializing functions which allocate resources from the shared memory mapped from an Intel® Xeon Phi™ coprocessor
- DCFA Modified IB core
- DCFA IB IF  
Provides an IB communication API for Intel® Xeon Phi™ coprocessor based applications

### Evaluation Scenario

- Intel® Xeon Phi™ coprocessor to Intel® Xeon Phi™ coprocessor data transfer using DCFA
- Host to host data transfer using the Infiniband Verb API

### Results of the Evaluation

- The same performance as that of host to host data transfer for large message sizes.



## Intel MPI for Intel® Xeon Phi™ coprocessors

### Target cluster environment

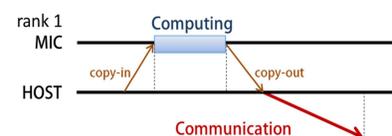
- A Linux OS runs on an Intel® Xeon Phi™ coprocessor or a host, Intel® Xeon Phi™ coprocessor based applications run above this Linux

### SCIF (Symmetric Communication Interface)

- Provides a mechanism for inter-node communication within a single platform
- A node is defined as either a host processor or an Intel® Xeon Phi™ coprocessor.
- Present the same interface on both the host processor and the Intel® Xeon Phi™ coprocessor.
- Provided from KNC's MPSS

### Programming Model

- Many-core Hosted (internode communication not implemented yet)
  - MPI ranks on Intel® Xeon Phi™ coprocessors only
- Symmetric
  - MPI ranks on Intel® Xeon Phi™ coprocessors and host processors.
  - Communication supported between 2 Intel® Xeon Phi™ coprocessors, 2 host processors, and between an Intel® Xeon Phi™ coprocessor and a host processor
- MPI + Offload
  - MPI ranks on host CPUs only
  - Offloads computing to Intel® Xeon Phi™ coprocessors to accelerate MPI ranks
    - Copy-in data before computing, and copy-out data before communication
    - Performs communication on the host, and performs computing on the Intel® Xeon Phi™ coprocessors.



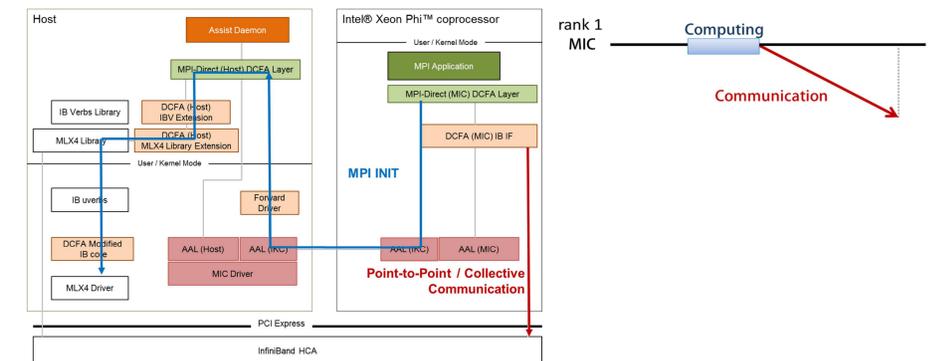
## DCFA-MPI

### Target cluster environment

- Intel® Xeon Phi™ coprocessors work as computing nodes, the Intel® Xeon Phi™ coprocessor based applications run in a **Lightweight Micro Kernel** but not above a Linux OS.

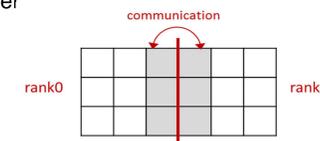
### Programming Model

- Initializes the Infiniband resource on the host
- Direct Point-to-Point / Collective communication on the Intel® Xeon Phi™ coprocessors.**
- Computing on the Intel® Xeon Phi™ coprocessors
- Offloads standard IO, File IO, communication using user-defined data types to host



### Evaluation

- Compared to the Intel MPI on Xeon + Offload to Pre-Production Intel® Xeon Phi™ coprocessors.
- Two-Dimensional Rectangle Stencil Computing with 2 Ranks
- Omits "vector computing" and only measures the data transfer



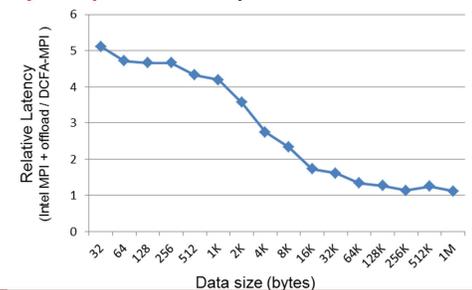
```
do {
    /* vector computing; */
    MPI_Isend( sbuf, n, MPI_INT, dst ... );
    MPI_Irecv( rbuf, n, MPI_INT, src ... );
    MPI_Waitall( 2, reqs, status );
} while ( ++t < TIME);
```

```
do {
    #pragma offload target (mic) ¶
    /* vector computing; */
    MPI_Isend( sbuf, n, MPI_INT, dst ... );
    MPI_Irecv( rbuf, n, MPI_INT, src ... );
    MPI_Waitall( 2, reqs, status );
} while ( ++t < TIME);
```

### Results of the Evaluation

- DCFA-MPI (MPI on Pre-Production Intel® Xeon Phi™ coprocessor) shows a better communication performance than the Intel MPI on Xeon + Offload to Pre-Production Intel® Xeon Phi™ coprocessors, achieves a **maximum speedup of 5x** at 32bytes.

Machine	Intel Workstation
CPU	Intel Xeon X5680 3.33GHz x 2
Infiniband HCA	Mellanox MT26428
Card	Pre-Production Intel® Xeon Phi™ coprocessor x 1



## Future Work

- Implementing intra-node (inside Intel® Xeon Phi™ coprocessor) communication using OpenMP
- Evaluation using the HPL Benchmark