Efficient Intra-node Communication on Intel MIC Clusters

Sreeram Potluri Akshay Venkatesh Devendar Bureddy

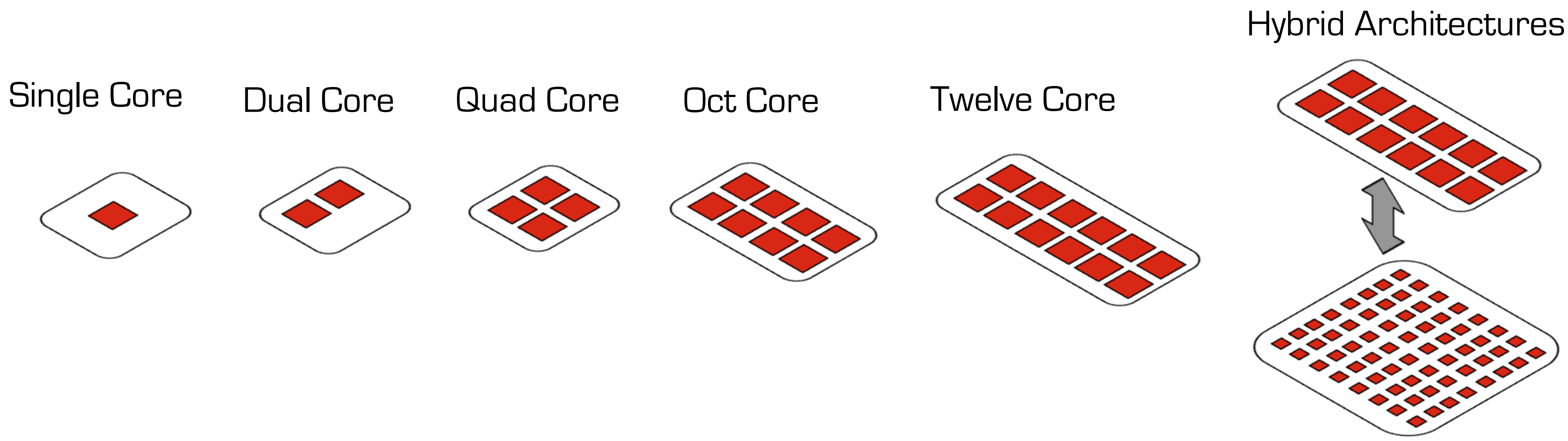
Krishna Kandalla Dhabaleswar K. Panda

Network-Based Computing Laboratory
Department of Computer Science and Engineering
The Ohio State University

Outline

- Introduction
- Problem Statement
- Hybrid MPI Communication Runtime
- Performance Evaluation
- Conclusion and Future Work

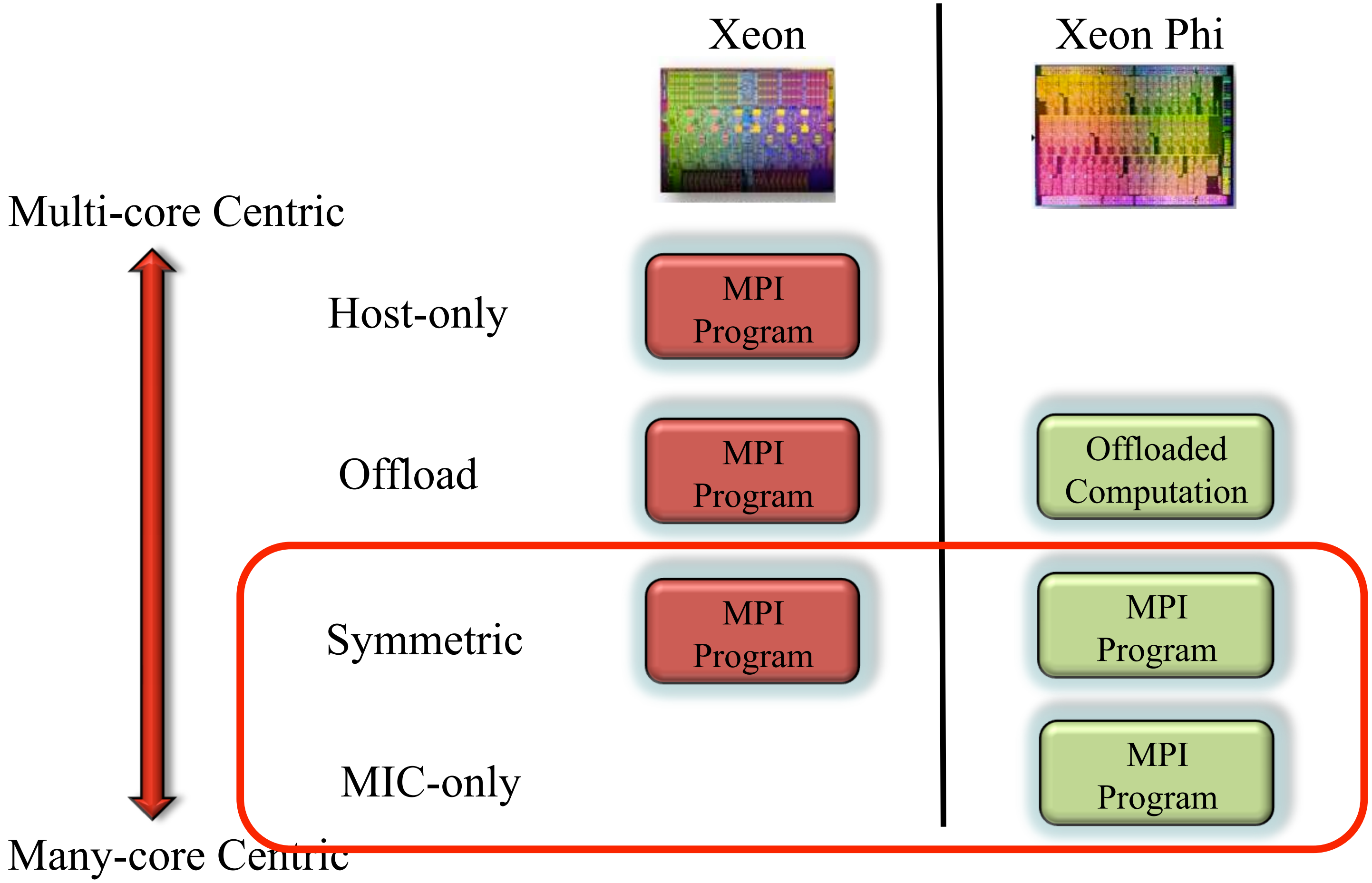
Many Integrated Core (MIC) Architecture



- Hybrid system architectures with graphics processors have become common - high compute density and high performance per watt
- Intel introduced Many Integrated Core (MIC) architecture geared for HPC
- X86 compatibility - applications and libraries can run out-of-the-box or with minor modifications
- Many low-power processor cores, hardware threads and wide vector units
- MPI continues to be a predominant programming model in HPC

NETWORK-BASED COMPUTING LABORATORY

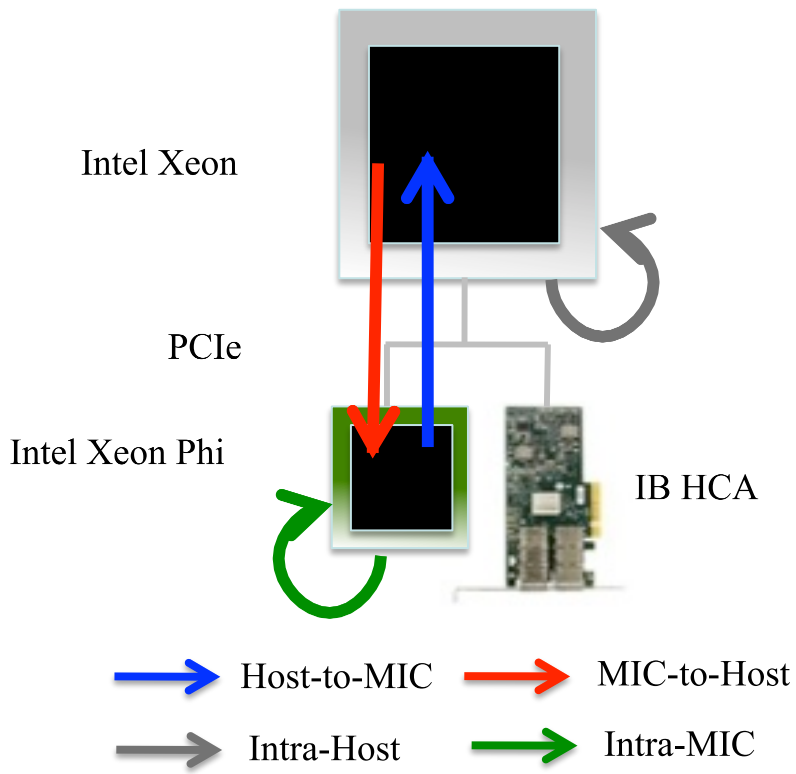
Programming Models on Clusters with MIC



- Xeon Phi, the first commercial product based on MIC architecture
- Flexibility in launching MPI jobs on clusters with Xeon Phi

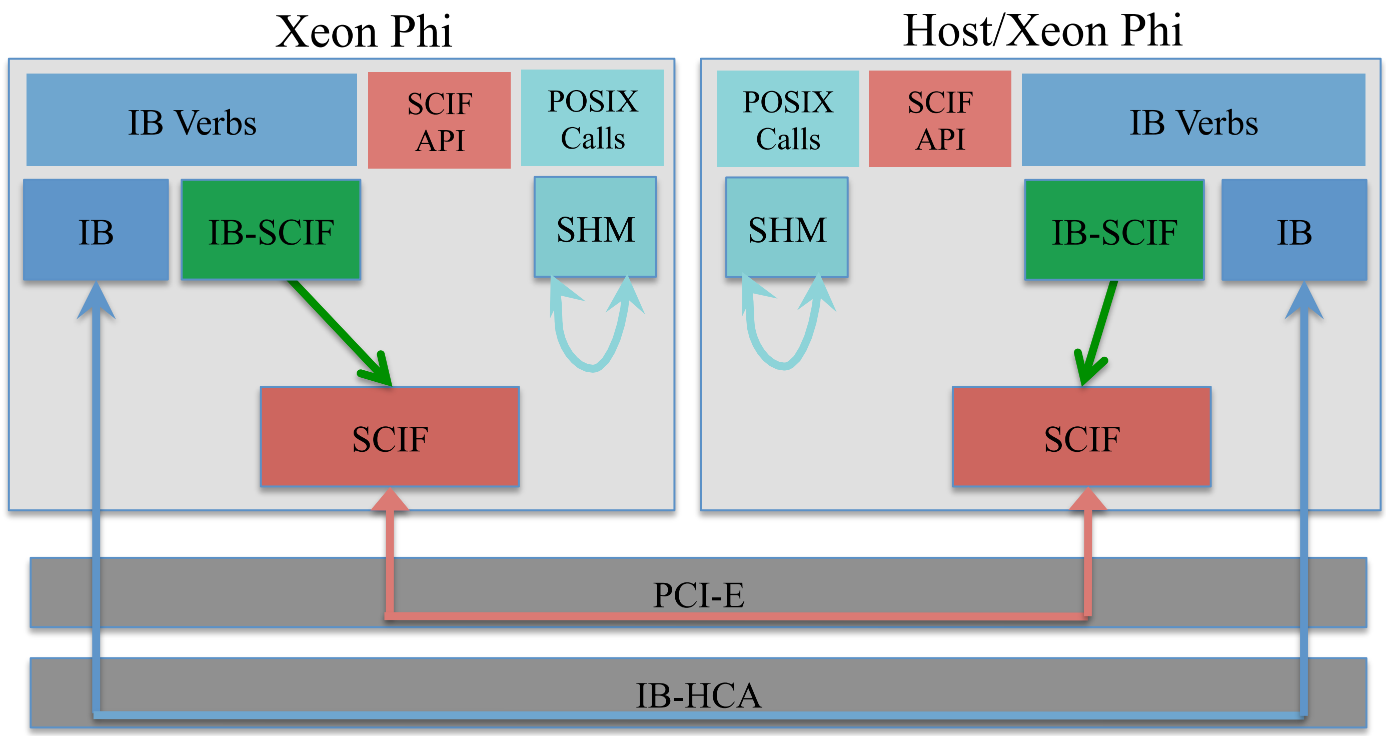
OHIO STATE

MPI Communication on Node with a Xeon Phi



- Various paths for MPI communication on a node with Xeon Phi

Symmetric Communication Stack with MPSS



- MPSS – Intel Manycore Platform Software Stack
  - Shared Memory
  - Symmetric Communication InterFace (SCIF) – over PCIe
  - IB Verbs – through IB adapter
  - IB-SCIF – IB Verbs over SCIF

Problem Statement

What are the performance characteristics of different communication channels available on a node with Xeon Phi?

How can an MPI communication runtime take advantage of the different channels?

Can a low latency and high bandwidth *hybrid communication channel* be designed, leveraging the all channels?

What is the impact of such a *hybrid communication channel* on performance of benchmarks and applications?

Outline

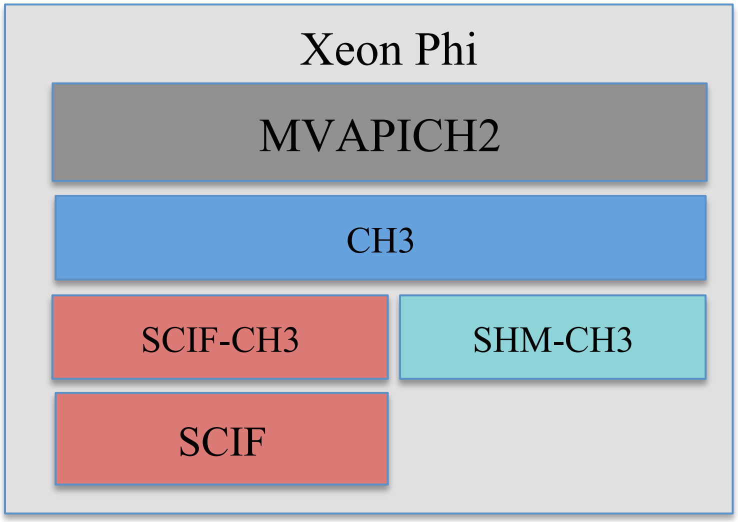
- Introduction
- Problem Statement
- Hybrid MPI Communication Runtime
- Performance Evaluation
- Conclusion and Future Work

MVAPICH2/MVAPICH2-X Software

- High Performance open-source MPI Library for InfiniBand, 10Gig/iWARP and RDMA over Converged Enhanced Ethernet (RoCE)
  - MVAPICH (MPI-1), MVAPICH2 (MPI-3.0), available since 2002
  - MVAPICH2-X (MPI + PGAS), Available since 2012
  - Used by more than 2,000 organizations (HPC Centers, Industry and Universities) in 70 countries
  - More than 165,000 downloads from OSU site directly
  - Empowering many TOP500 clusters
    - 7th ranked 204,900-core cluster (Stampede) at TACC
    - 14th ranked 125,980-core cluster (Pleiades) at NASA
    - and many others
  - Available with software stacks of many IB, HSE and server vendors including Linux Distros (RedHat and SuSE)
  - http://mvapich.cse.ohio-state.edu
- Partner in the U.S. NSF-TACC Stampede (9 PFlop) System

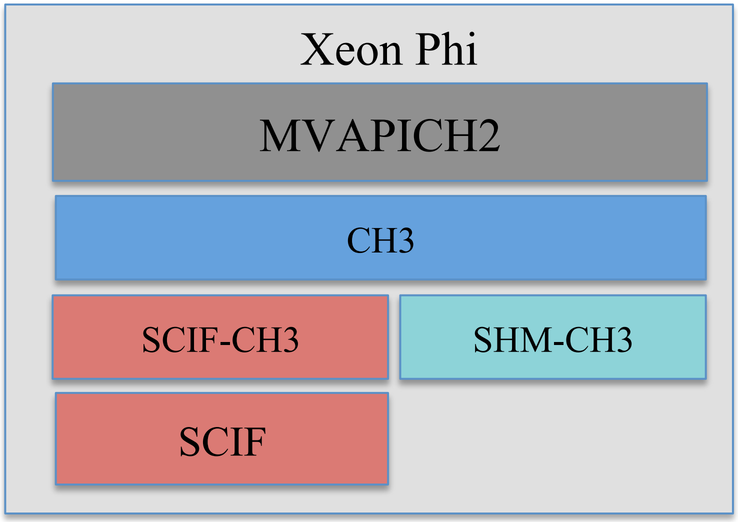
Intra-MIC Communication

- Shared Memory Interface (CH3-SHM)
  - POSIX Shared Memory API
  - Small Messages: pair-wise memory regions between processes
  - Large Messages: buffer pool per process, data is divided into chunks (8KB) to pipeline copy in and copy out
  - MPSS offers two implementations of *memcpy*
    - multi-threaded copy
    - DMA-assisted copy: offers low latency for large messages
  - We use 64KB chunks to trigger the use of DMA-assisted copies for large messages

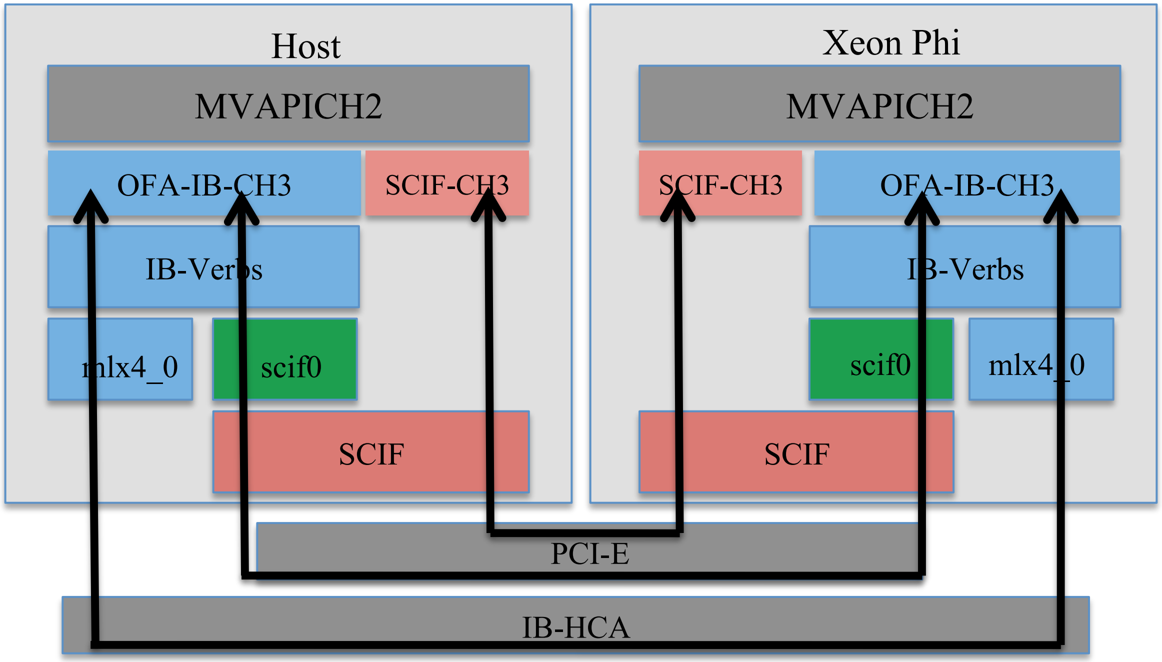


Intra-MIC Communication

- SCIF Channel (CH3-SCIF)
  - Control of DMA engine to the user
  - API for remote memory access:
    - Registration: scif_register
    - Initiation: scif_writeto/readfrom
    - Completion: scif_fence_signal
  - We use a write-based rendezvous protocol
    - Sender sends *Request-To-Send (RTS)*
    - Receiver responds with *Ready-to-Receive (RTR)* with registered buffer offset and flag offset
    - Sender issues *scif_writeto* followed by *scif_fence_signal*
    - Both processes poll for flag to be set



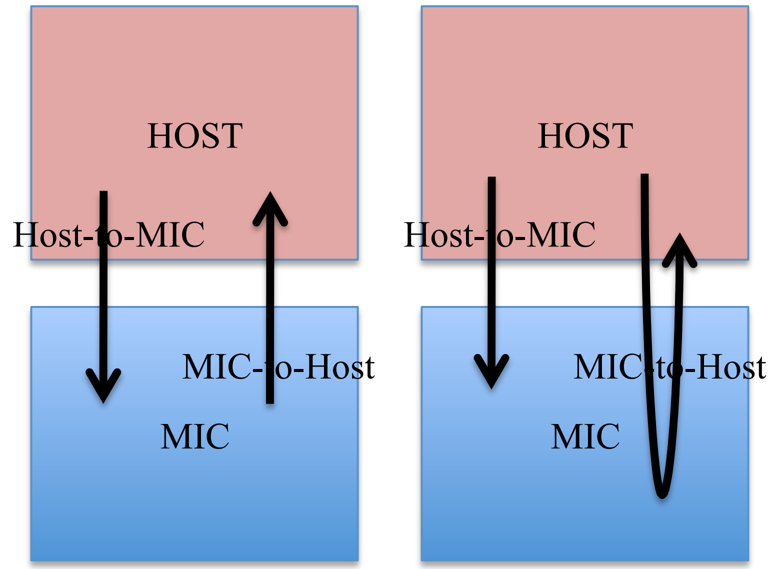
Host-MIC Communication



- IB Channel (OFA-IB-CH3)
  - Uses IB verbs
  - Selection of IB network interface to switch between IB and IB-SCIF
- SCIF-CH3
  - Can be used for communication between Xeon Phi and Host

Host-MIC Communication: Host-Initiated SCIF

- DMA can be initiated by host or Xeon Phi
- But performance is not symmetric
- Host-initiated DMA delivers better performance
- Host-initiated mode takes advantage of this
  - Write-based from Host-to-Xeon Phi
  - Read-based transfer from Xeon Phi-to-Host



Symmetric   Host-Initiated

- Symmetric mode to maximize resource utilization on host and Xeon Phi

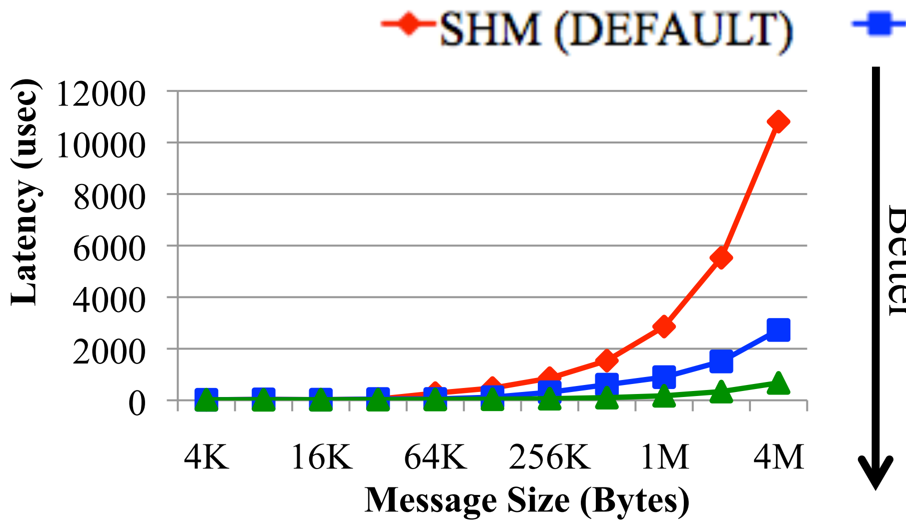
Outline

- Introduction
- Problem Statement
- Hybrid MPI Communication Runtime
- Performance Evaluation
- Conclusion and Future Work

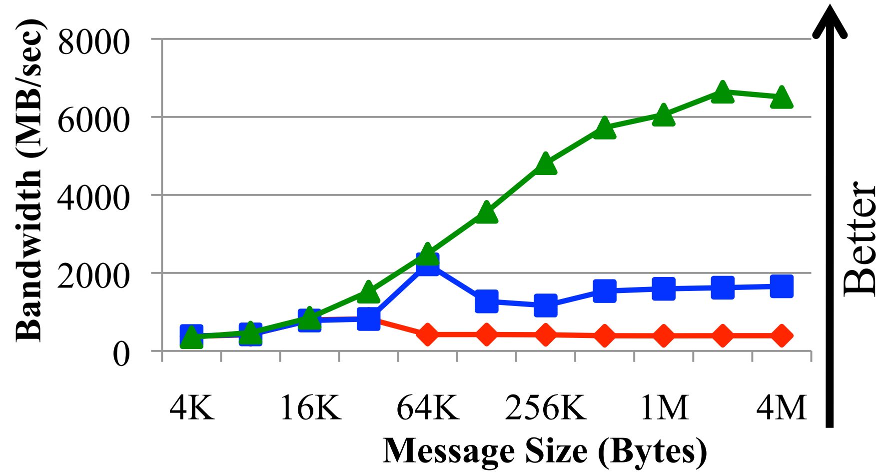
Experimental Setup

- TACC Stampede Node
  - Host
    - Dual-socket oct-core Intel Sandy Bridge (E5-2680 @ 2.70GHz)
    - CentOS release 6.3 (Final)
  - MIC
    - SE10P (B0-KNC)
    - 61 cores @ 1085.854 MHz, 4 hardware threads/core
    - OS 2.6.32-279.el6.x86_64, MPSS 2.1.4346-16
  - Compiler: Intel Composer_xe_2013.2.146
  - Network Adapter: IB FDR MT 4099 HCA
  - Enhanced MPI based on MVAPICH2 1.9

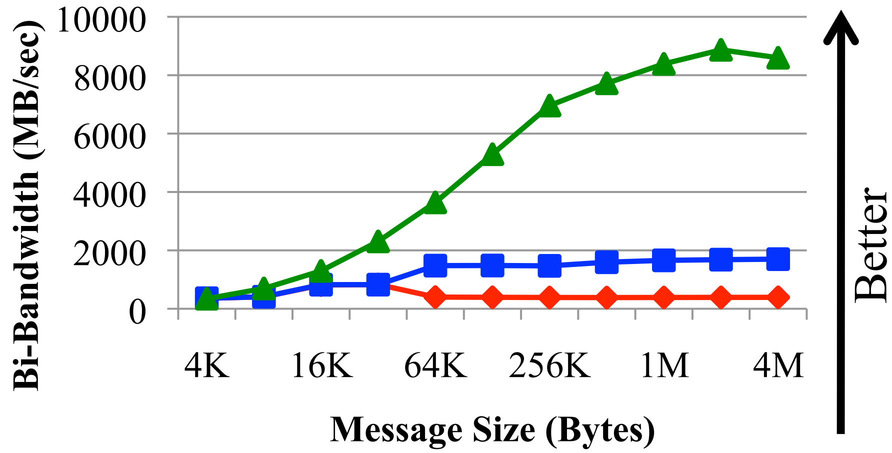
Intra-MIC Point-to-Point Communication



osu_latency



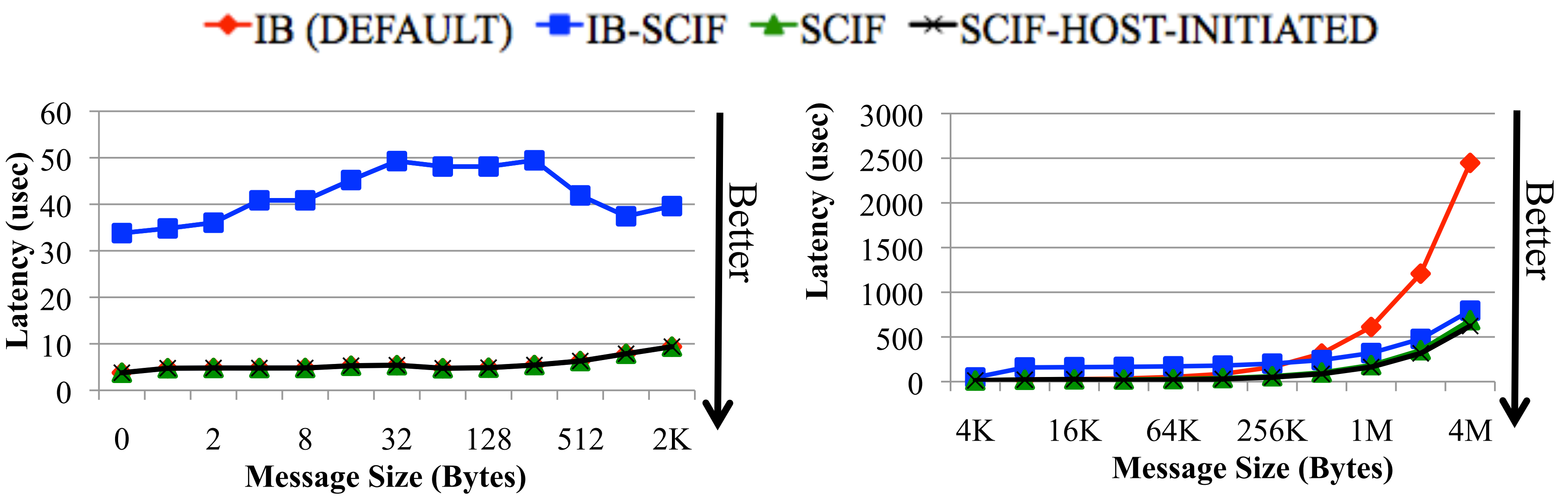
osu_bw



osu_bibw

- Default chunk size severely limits performance
- Tuned block size alleviates it but shm performance still low
- Using SCIF works around these limitations – 75% improvement in latency, 4.0x improvement in b/w over SHM-TUNED

Host-MIC Point-to-Point Communication



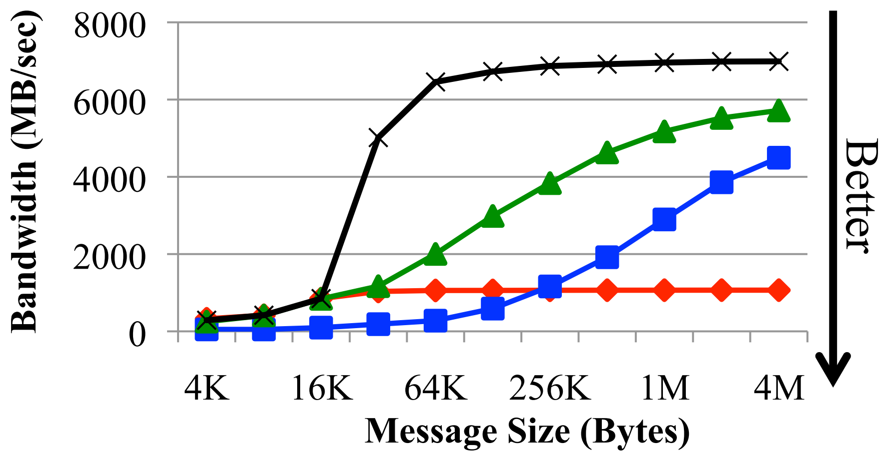
osu_latency : small messages

osu_latency : large messages

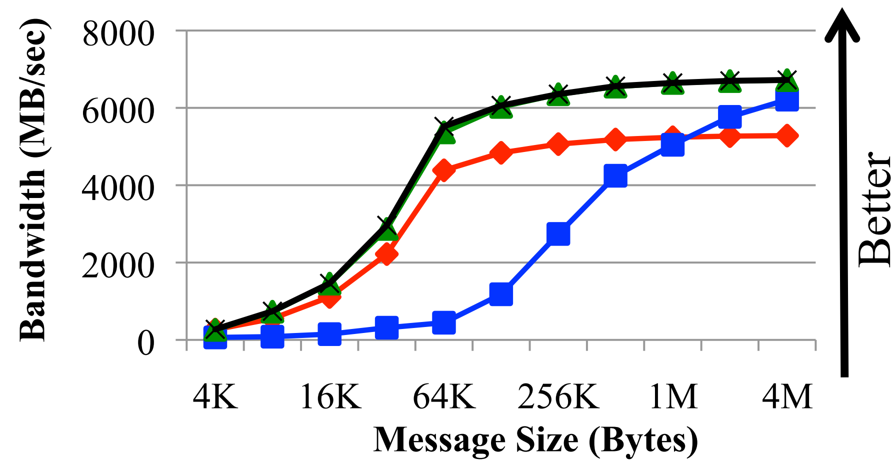
- IB provides a low-latency path – 4.7usec for 4Byte messages
- IB-SCIF overheads due to SCIF and additional software layer
- SCIF designs are already hybrid, use IB for small messages
- SCIF outperforms IB for large messages – 72% improvement for 4MB messages
- Host-Initiated SCIF takes advantage of faster DMA – 33% improvement over SCIF for 64KB messages

Host-MIC Point-to-Point Communication

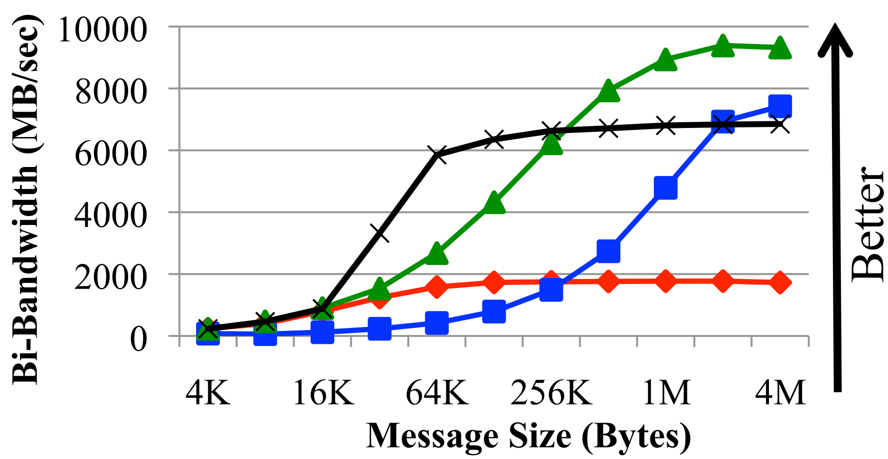
IB (DEFAULT) — IB-SCIF — SCIF — SCIF-HOST-INITIATED



osu_bw: mic-to-host



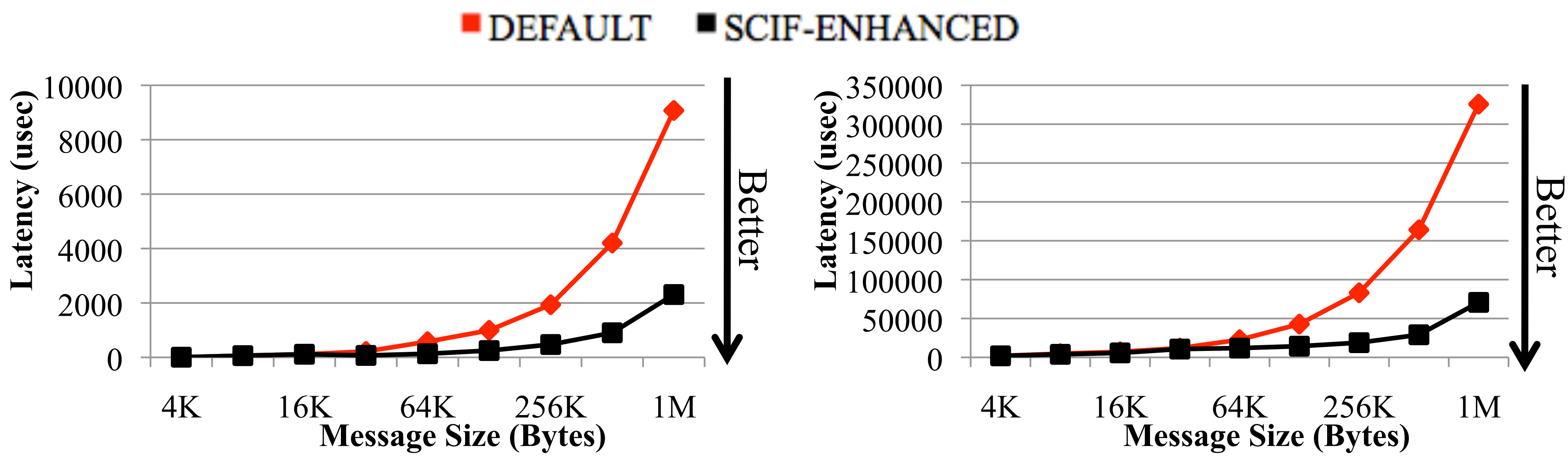
osu_bw: host-to-mic



osu_bibw

- IB bandwidth limited mic-to-host due to peer-to-peer limitation on Sandy Bridge
- SCIF works around this, Host-initiated DMA delivers better bandwidth too – 6.6x improvement over IB
- Host-initiated SCIF worse than SCIF in bibw due to wasted resources

Collective Communication

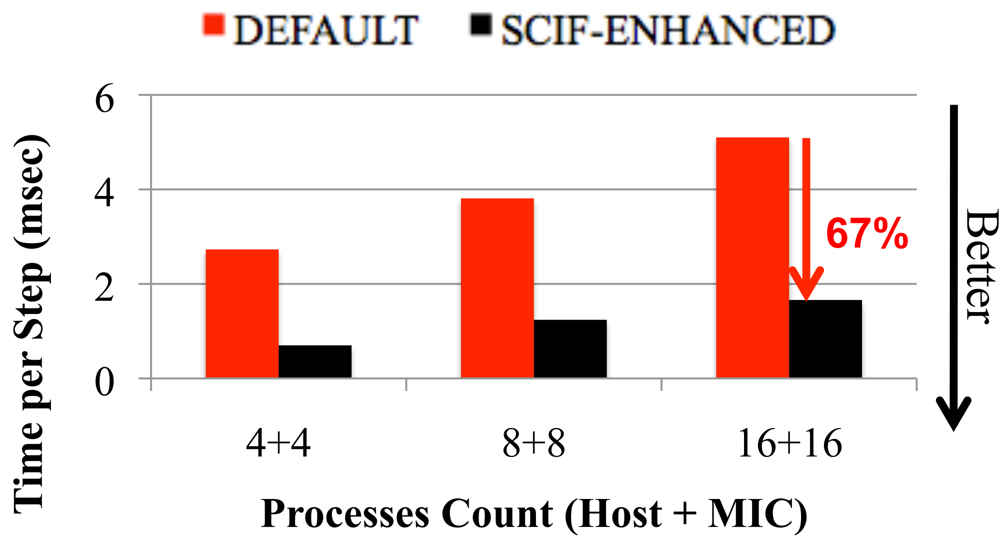


osu_gather: root on host

osu_alltoall

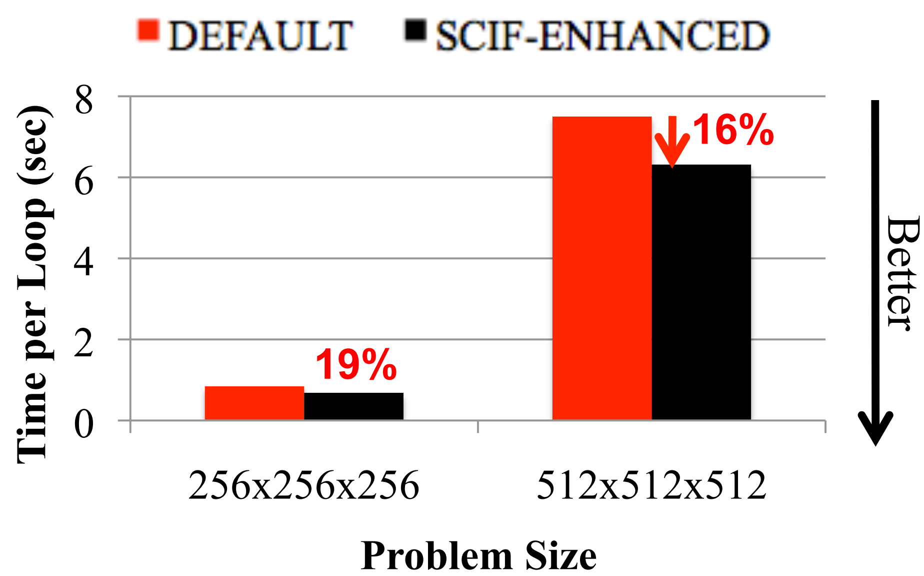
- 16 processes on host + 16 processes on MIC
- Host-initiated SCIF or symmetric SCIF based on the communication pattern and message size, collective level selected
- Gather, rooted collective uses host-initiated SCIF – 75% improvement in at 1MB
- All-to-all uses symmetric SCIF – 78% improvement at 1MB

Performance of 3D Stencil Communication Benchmark



- Near-neighbor communication – upto 6 neighbors – 64KB messages
- 67% improvement in time per step

Performance of P3DFFT Library



- (MPI + OpenMP) version of popular library for 3D Fast Fourier Transforms - test performs forward transform and a backward transform in each iteration
- 2 processes on Host (8 threads/process) + 8 processes on MIC (8 threads/process)
- Uses symmetric SCIF because of the MPI_Alltoall
- Upto 19% improvement using SCIF-ENHANCED

Conclusion and Future Work

- A hybrid communication runtime to optimize intranode MPI communication on clusters with Xeon Phi
- Take advantage of SCIF in addition to standard channels like shared memory and IB
- Upto 75% improvement in latency and 6x improvement in unidirectional bandwidth for MIC-Host Communication
- Upto 78% improvement in MPI_Alltoall performance
- Considerable improvements with 3DStencil and P3DFFT kernels
- Focus on optimizations for shared memory based communication
- Working on designs for inter-node communication on clusters with Xeon Phi

Thank You!

{potluri, akshay, bureddy, kandalla, panda} @cse.ohio-state.edu

Network-Based Computing Laboratory

http://nowlab.cse.ohio-state.edu/

MVAPICH Web Page

http://mvapich.cse.ohio-state.edu/