MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

OpenSHMEM NonBlocking Data Movement Operations with MVAPICH2-X: Early Experiences

PGAS Applications Workshop (PAW'16)

by

Khaled Hamidouche*, Jie Zhang*, Karen Tomko+, D.K Panda*

*: The Ohio State University (OSU), +: Ohio Supercomputing Center (OSC)

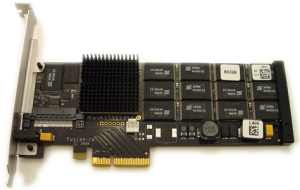
E-mail: hamidouc@cse.ohio-state.edu

Drivers of Modern HPC Cluster Architectures

Multi-core Processors

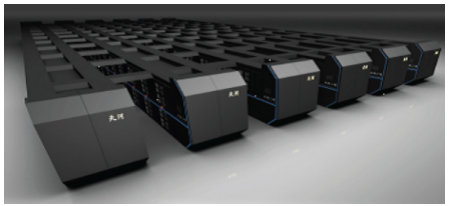
High Performance Interconnects - InfiniBand
<1usec latency, 200Gbps Bandwidth>

Accelerators / Coprocessors
high compute density, high performance/watt
>1 TFlop DP on a chip



SSD, NVMe-SSD, NVRAM

- Multi-core/many-core technologies
- Remote Direct Memory Access (RDMA)-enabled networking (InfiniBand and RoCE)
- Solid State Drives (SSDs), Non-Volatile Random-Access Memory (NVRAM), NVMe-SSD
- Accelerators (NVIDIA GPGPUs and Intel Xeon Phi)



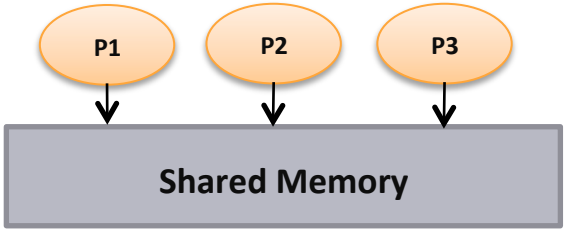
Tianhe – 2

Titan

Stampede

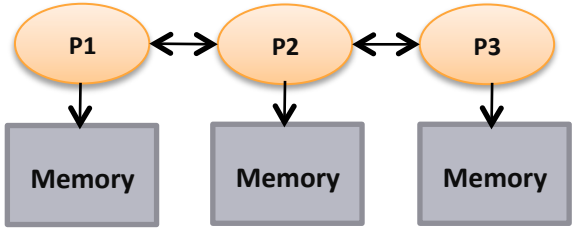
Tianhe – 1A

Parallel Programming Models Overview



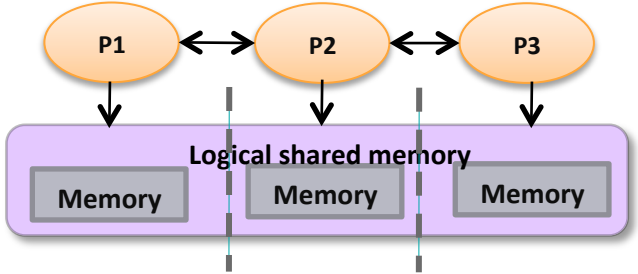
Shared Memory Model

DSM



Distributed Memory Model

MPI (Message Passing Interface)



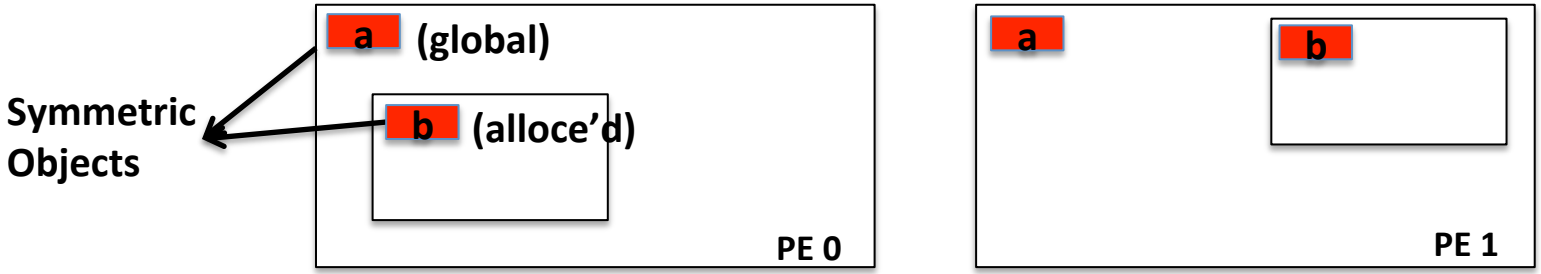
Partitioned Global Address Space (PGAS)

OpenSHMEM, GA, UPC, Chapel, X10, CAF, …

- Programming models provide abstract machine models
- Models can be mapped on different types of systems
  - e.g. Distributed Shared Memory (DSM), MPI within a node, etc.
- Additionally, OpenMP can be used to parallelize computation within the node
- Each model has strengths and drawbacks - suite different problems or applications

The OpenSHMEM Memory Model

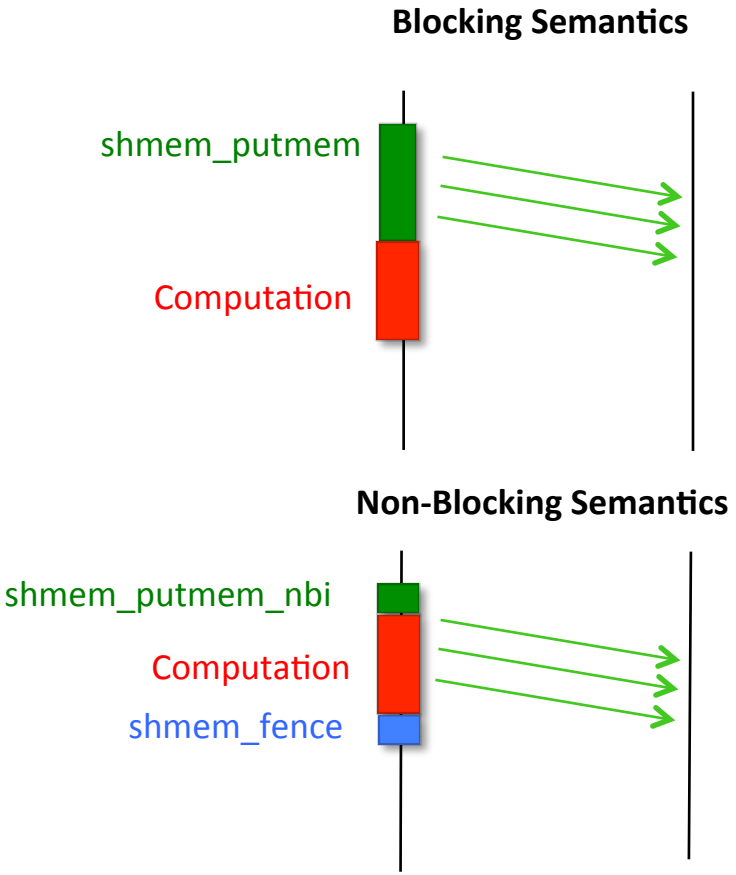
- Symmetric data objects
  - Global Variables
  - Allocated using collective *shmem_malloc, shmem_memalign, shmem_realloc* routine



- Globally addressable – objects have same
  - Type
  - Size
  - Same virtual address or offset at all PEs
  - Address of a remote object can be calculated based on info of local object
  - OpenSHMEM 1.3 introduces Non-Blocking Data Movement Operations

Non-Blocking Data Movement Operations

- Blocking operation => Optimize for latency
  - Buffer reuse after returning from the call
- Non-Blocking operation => Optimize Computation/ Communication overlap
  - Return as soon as we post the request
  - Completion is ensured later on a completion/ synchronization call
  - Buffer reuse after completion
  - API extension with _nbi (ex: shmem_putmem_nbi)
  - shmem_fence/Shmem_barrier. Complete all previous operations



Outline

- Introduction
- Contributions
- Alternative Designs
- Performance Evaluation
- Conclusions and Future Work

Contributions

- Propose high-performance designs and implementations of OpenSHMEM NBI operations on top of the MVAPICH2-X library.
- Extend OMB with new NBI benchmarks for evaluating OpenSHMEM 1.3 NBI operations in a standardized manner.
- Design communication kernels including 3D stencil and alltoall patterns using OpenSHMEM.
- Demonstrate the benefits and impact of OpenSHMEM NBI operations on both latency and overlap metrics.

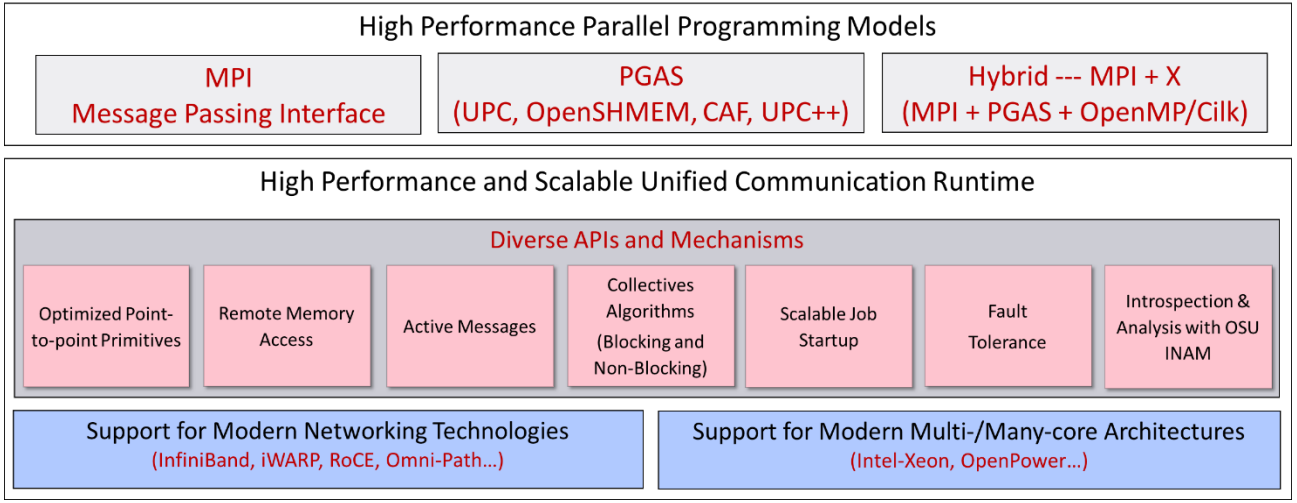
Outline

- Introduction
- Contributions
- Alternative Designs
- Performance Evaluation
- Conclusions and Future Work

Overview of the MVAPICH2 Project

- High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.0), Started in 2001, First version available in 2002
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
  - Support for Virtualization (MVAPICH2-Virt), Available since 2015
  - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
  - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) since 2015
  - **Used by more than 2,675 organizations in 83 countries**
  - **More than 399,000 (> 0.39 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Jun '16 ranking)
    - 12th ranked 519,640-core cluster (Stampede) at TACC
    - 15th ranked 185,344-core cluster (Pleiades) at NASA
    - 31st ranked 76,032-core cluster (Tsubame 2.5) at Tokyo Institute of Technology and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat and SuSE)
  - http://mvapich.cse.ohio-state.edu
- Empowering Top500 systems for over a decade
  - System-X from Virginia Tech (3rd in Nov 2003, 2,200 processors, 12.25 TFlops) ->
  - Stampede at TACC (12th in Jun'16, 462,462 cores, 5.168 Plops)

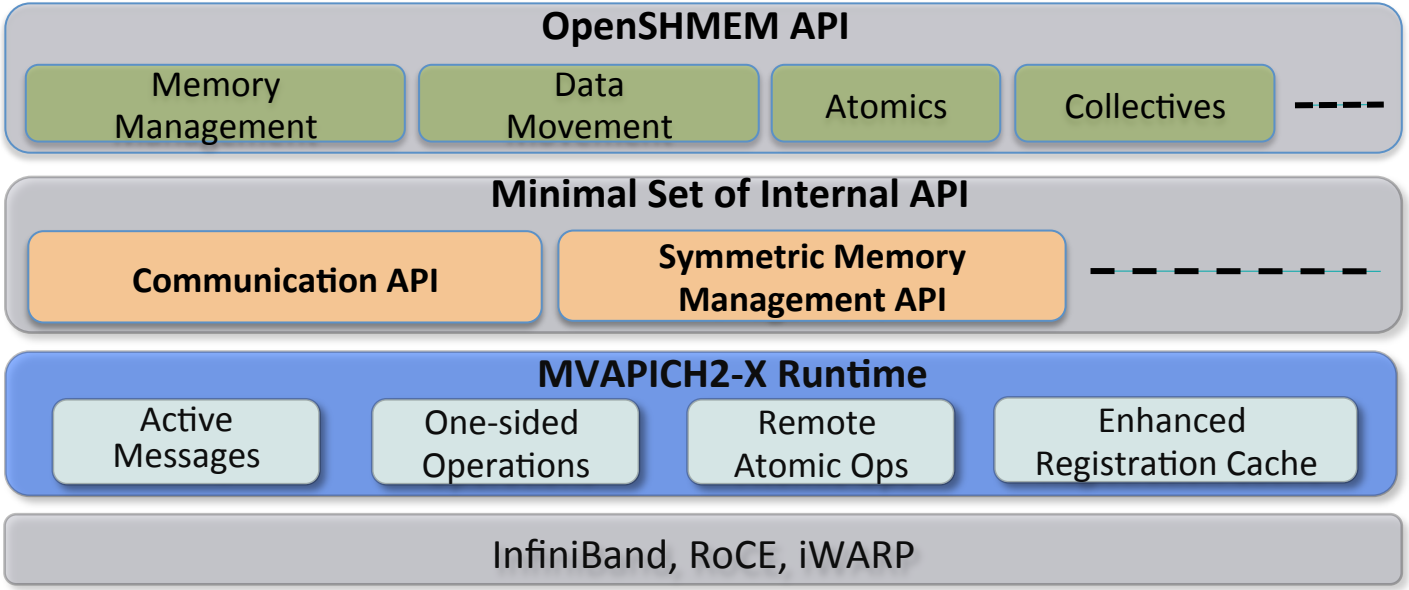
MVAPICH2-X for Hybrid MPI + PGAS Applications



- Unified communication runtime for MPI, UPC, UPC++, OpenSHMEM, CAF
  - Available since 2012 (starting with MVAPICH2-X 1.9)
  - http://mvapich.cse.ohio-state.edu

- Feature Highlights
  - Supports MPI(+OpenMP), OpenSHMEM, UPC, CAF, UPC++, MPI(+OpenMP) + OpenSHMEM, MPI(+OpenMP) + UPC
  - MPI-3 compliant, OpenSHMEM v1.3 standard compliant, UPC v1.2 standard compliant (with initial support for UPC 1.3), CAF 2008 standard (OpenUH), UPC++
  - Scalable Inter-node and intra-node communication – point-to-point and collectives

OpenSHMEM Design in MVAPICH2-X



- OpenSHMEM Stack based on OpenSHMEM Reference Implementation
- OpenSHMEM Communication over MVAPICH2-X Runtime
  - Uses active messages, atomic and one-sided operations and remote registration cache

J. Jose, K. Kandalla, M. Luo and D. K. Panda, Supporting Hybrid MPI and OpenSHMEM over InfiniBand: Design and Performance Evaluation, Int'l Conference on Parallel Processing (ICPP '12), September 2012

Intra-node Alternative Designs

- Shared Memory based
  - Symmetric Memory (heap) in shared memory
  - Direct copy from source to destination
  - Latency optimized (no overlap)
- CMA based
  - Shared memory is not available
  - Direct copy (Zero-copy)
  - Latency optimized (no overlap)
- IB Loopback based
  - Offload the copy operations to an external engine (IB)
  - Exchange the l_key/r-key during initialization
  - Overlap optimized (return as soon as we post the IB operation)
- On-load engine based (Work in progress)
  - Kernel based helper threads
  - Offload the copy operations to helper threads (Similar to CMA)
  - Optimized for both Latency and Overlap

Inter-nodes Alternative Designs

- List-based design
  - On the call: Create and en-queue an internal request
    - Associate an IB Completion Event with a request
  - In the progress engine: De-queue and delete a completed request
  - During completion call (Fence): Polls the progress engine until the list is empty
  - Overhead of Create/Delete of the request in Critical path
- Counter-based design
  - Global integer counter
  - On the call: Increment the counter
  - In the progress engine: decrease the counter
  - During completion call: Poll the progress engine until counter==0
  - Minimal overhead in the critical path

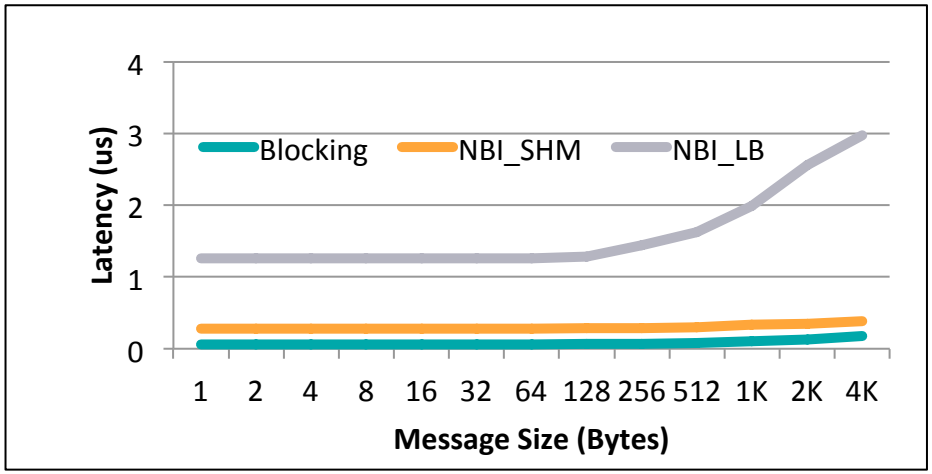
Outline

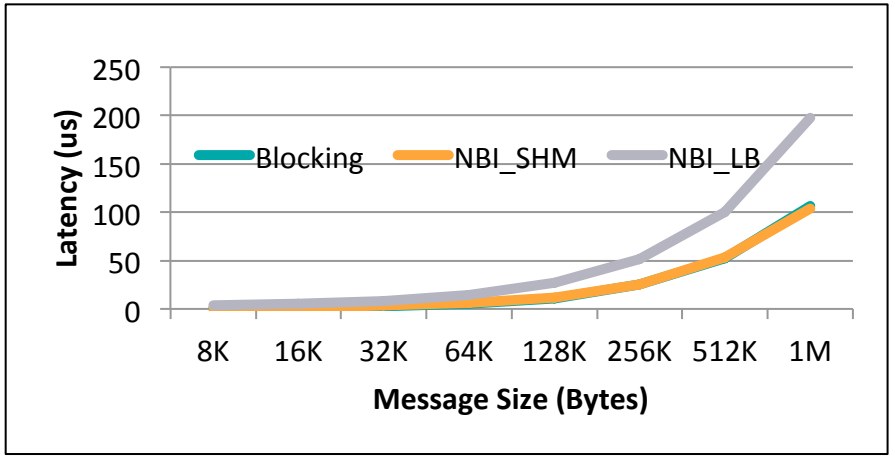
- Introduction
- Contributions
- Alternative Designs
- Performance Evaluation
- Conclusions and Future Work

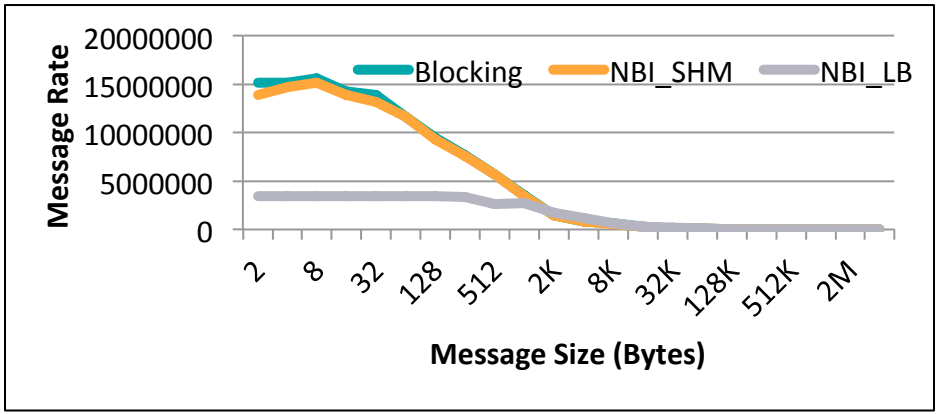
Experimental Setup

- Stampede System @TACC
  - 16-cores Sandy Bridge nodes
  - Mellanox FDR interconnection
- MVAPICH2-X 2.2RC1
  - Unified Communication Runtime support (UCR)
  - OpenSHMEM 1.3 git branch of the standard implementation
- Extension of OpenSHMEM OMB with
  - NBI pt-pt latency tests (for put and get)
  - Message rate benchmarks
  - Overlap Benchmarks
- Redesigned All-to-All and 3D-Stencil benchmarks with NBI interface

Intra-node Evaluation

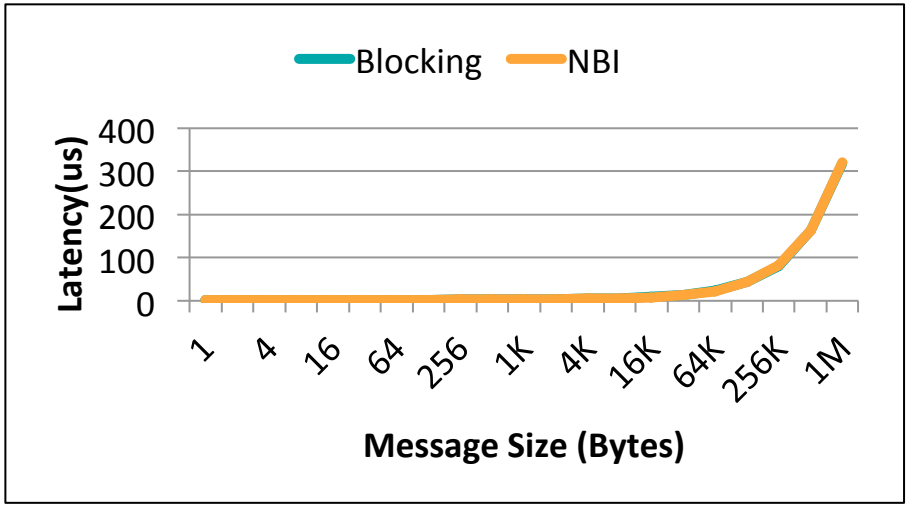


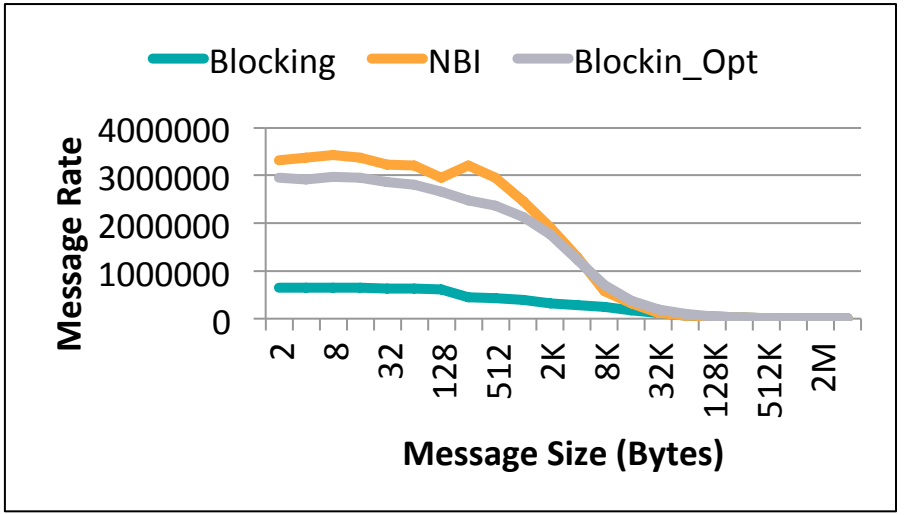


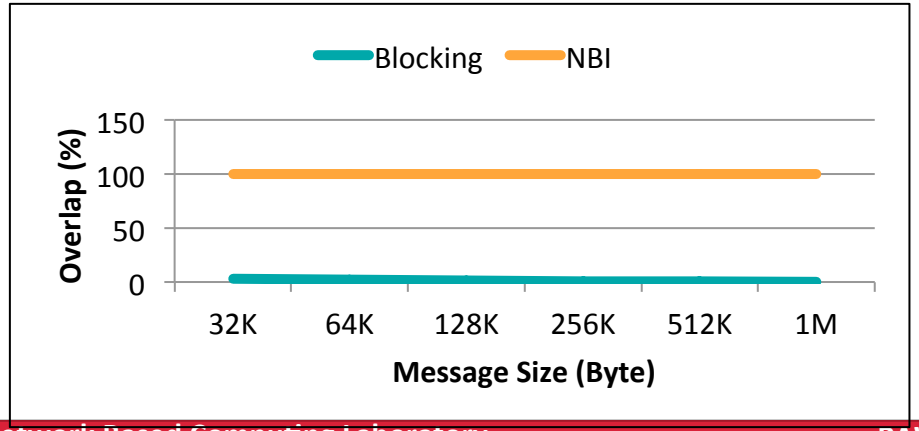


- LB-based design has overhead in latency
- SHM-based design achieves very good message rate (3X) improvements compared to LB-design
- Overhead in latency for small message
  - Software overhead due to synchronization operation

Inter-node Evaluation

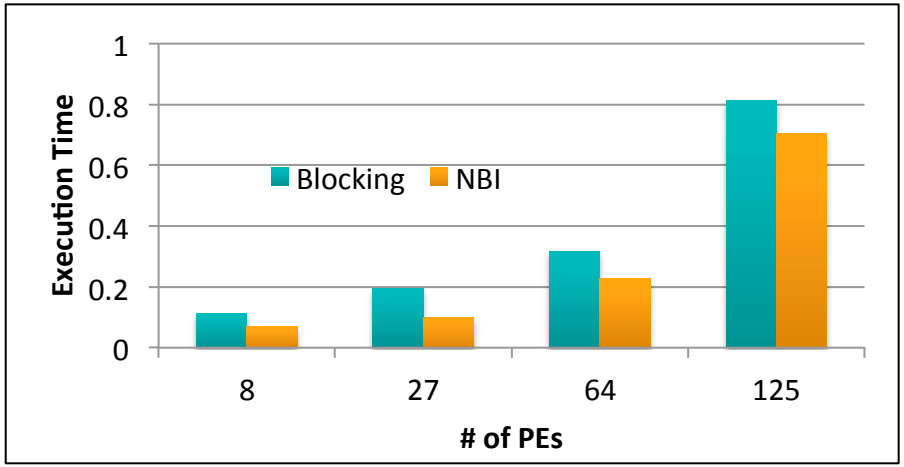




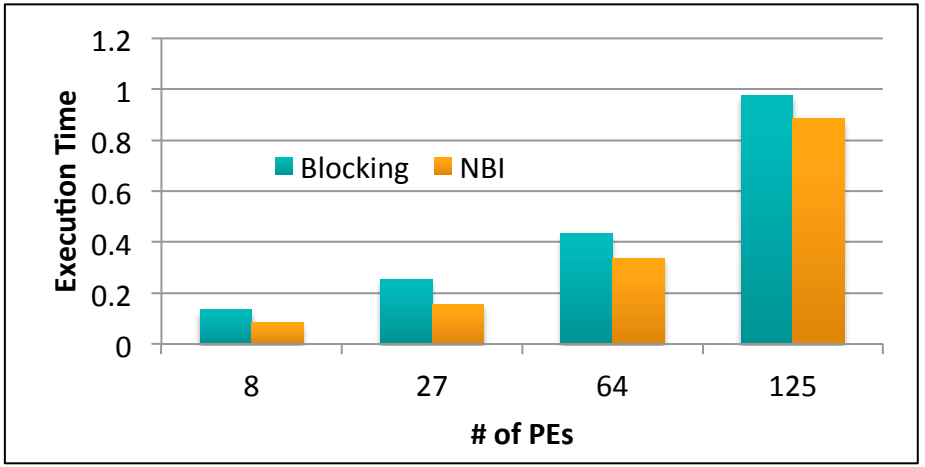


- NBI delivers same latency performance as Blocking
- NBI design achieves very good message rate (5X) compared to blocking
- Maximal overlap potential
  - Hide the communication overhead with RDMA

3D-Stencil Communication Kernel Evaluation



Small Input Size (512 Byte)



Large Input Size (2 KByte)

- 50%, 30% and 15% performance improvement on 27, 64 and 125 cores
- Overlap benefits
  - Small message: both computation/Communication and Communication/Communication overlap

Outline

- Introduction
- Contributions
- Alternative Designs
- Performance Evaluation
- Conclusions and Future Work

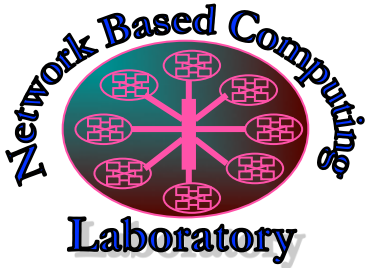
Conclusions

- Highlighted the alternative approaches for designing NBI operations
  - Both intra and internode operation on RDMA networks
- Demonstrate the benefits of OpenSHMEM NBI operations
  - Message Rate, Overlap

- Evaluate the impact of such semantics/designs at application level

- The support will be available with the next release of MVAPICH2-X
- The new Benchmarks will be available with the next release of OMB

Thank You!



Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu/

The MVAPICH Project
http://mvapich.cse.ohio-state.edu/

hamidouc@cse.ohio-state.edu