NETWORK-BASED COMPUTING LABORATORY

High Performance Design and Implementation of Nemesis Communication Layer for Two-sided and One-Sided MPI Semantics in MVAPICH2

Miao Luo, Sreeram Potluri, Ping Lai, Emilio P. Mancini, Hari Subramoni, Krishna Kandalla, Sayantan Sur, D. K. Panda

Network-based Computing Lab

The Ohio State University

OHIO STATE

Outline

- Introduction & Motivation
- Problem Statement
- Design Challenges
- Evaluation of Performance
- Conclusions and Future Work

Introduction

- Message Passing Interface
  - Pre-dominant parallel programming model
  - Deployed by many scientific applications
    - *Earthquake Simulation*
    - *Weather prediction*
    - *Computational Fluid dynamics*
    - *…*

Introduction

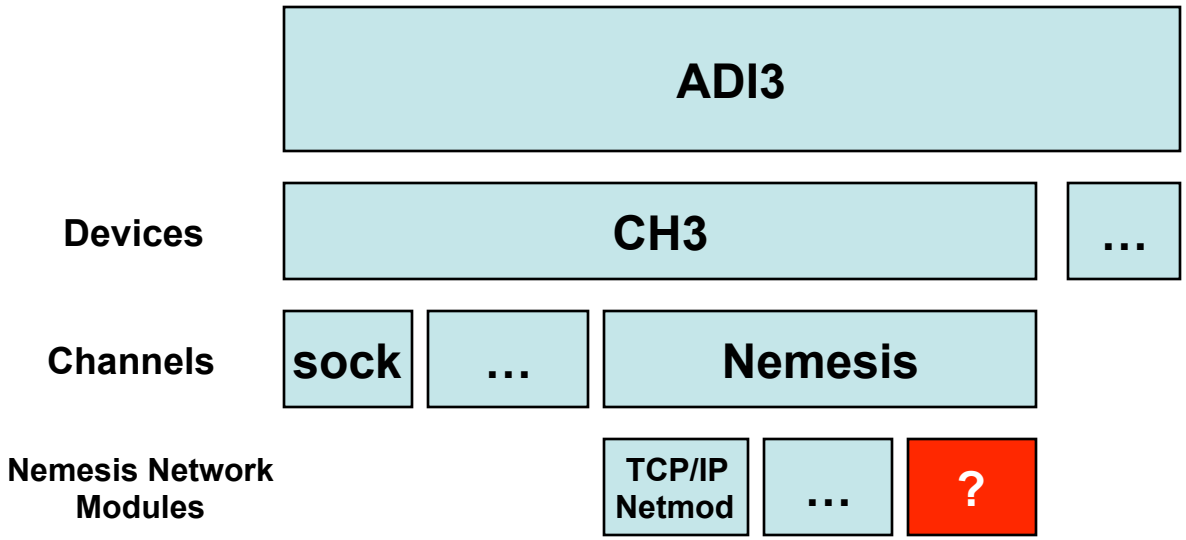
- MPI-2 R(emote) M(emory) A(ccess)
  - Allow one process involved in data transfer.
  - Data transfer operations:
    - *MPI_Put*
    - *MPI_Get*
    - *MPI_Accumulate*
  - Synchronization operations:
    - *Fence*
    - *Post-start-wait-complete*
    - *Lock/unlock*

Introduction

- MPICH2
  - Freely available, open-source, widely portable implementation of MPI standard
  - Re-designed for multi-core systems
  - Nemesis Communication Layer
    - Optimized for fast intra-node communication
      - Lock-free queues with shared memory
      - Kernel-based: KNEM
    - Modular design for various high-performance interconnects

Nemesis Communication Layer

- Nemesis Communication Layer
  - For scalability, high-performance intra-node communication
  - Modular design: multiple network modules
  - Envision: next generation and highest performing design for MPICH2

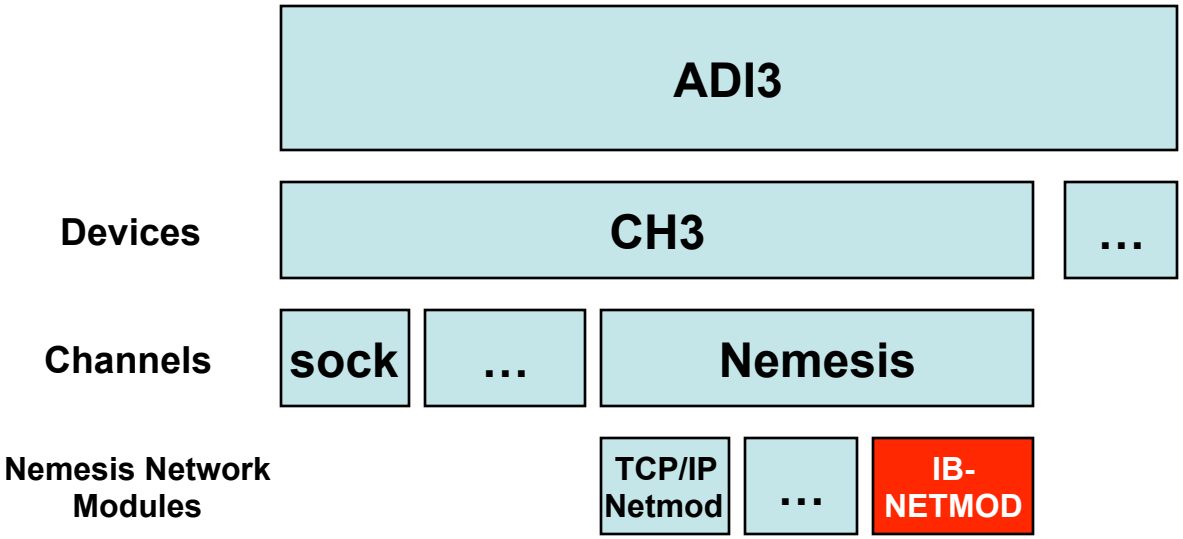


An overview of InfiniBand

- InfiniBand
  - High-speed, general purpose I/O interconnect
  - Widely used by scientific computing centers world-wide
  - 40% systems in Top500 (June 2010)
  - Two communication semantics
    - Channel semantics: send/recv
    - Memory semantics: RDMA

Motivation

- Nemesis + InfiniBand ?
- InfiniBand network module (IB-Netmod)
  - Expose InfiniBand's high-performance ability to intra-node optimized Nemesis Communication Layer



Outline

- Introduction
- Problem Statement
- Design Challenges
- Evaluation of Performance
- Conclusions and Future Work

Problem Statement

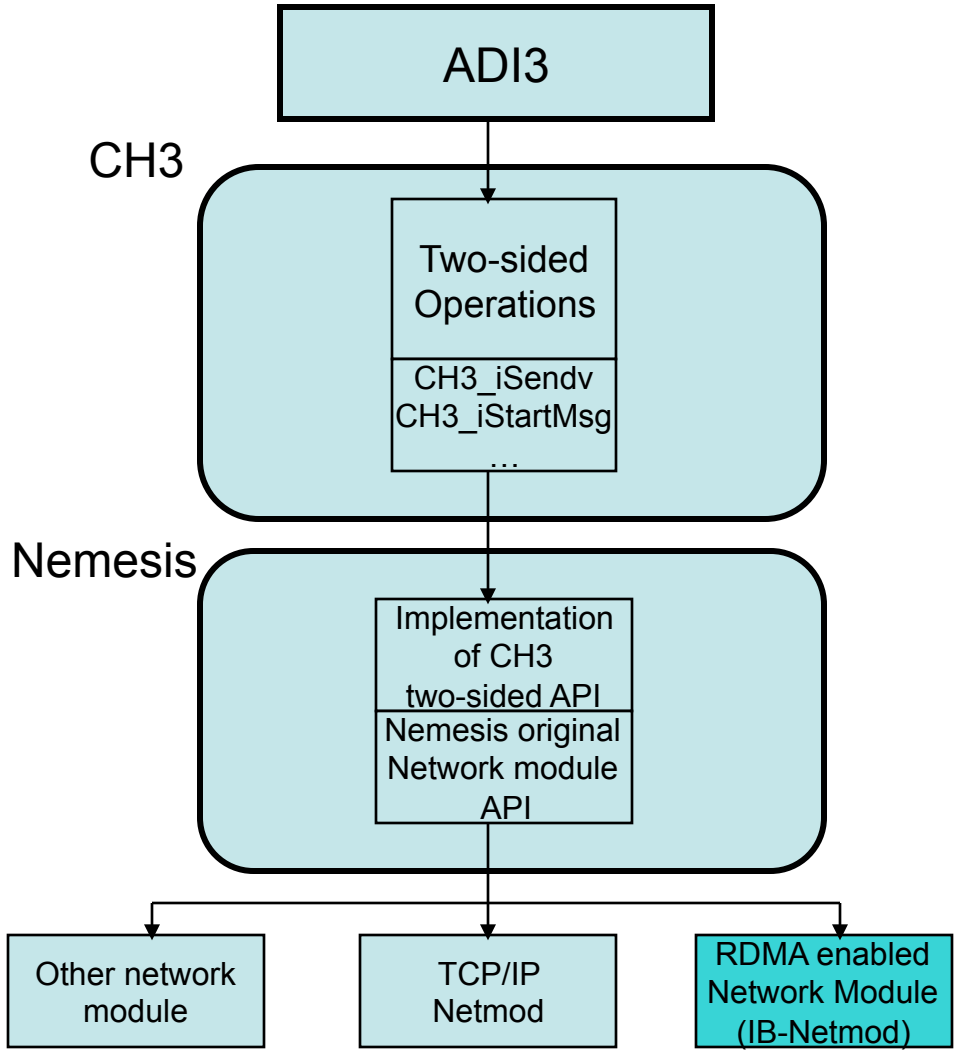
- What are the considerations for a high-performance network module?
  - Best **two-sided** performance
  - Efficiently utilize the **full ability** of interconnects
- **Limitation** of current ch3 and nemesis general API:
  - Can **extensions** be made to current layering API?
  - RMA functionality can be optimized by lower layer
- **Better performance** from extended Nemesis interface ?
  - while also keeping an **unified design**?
  - providing **modularity**?

Outline

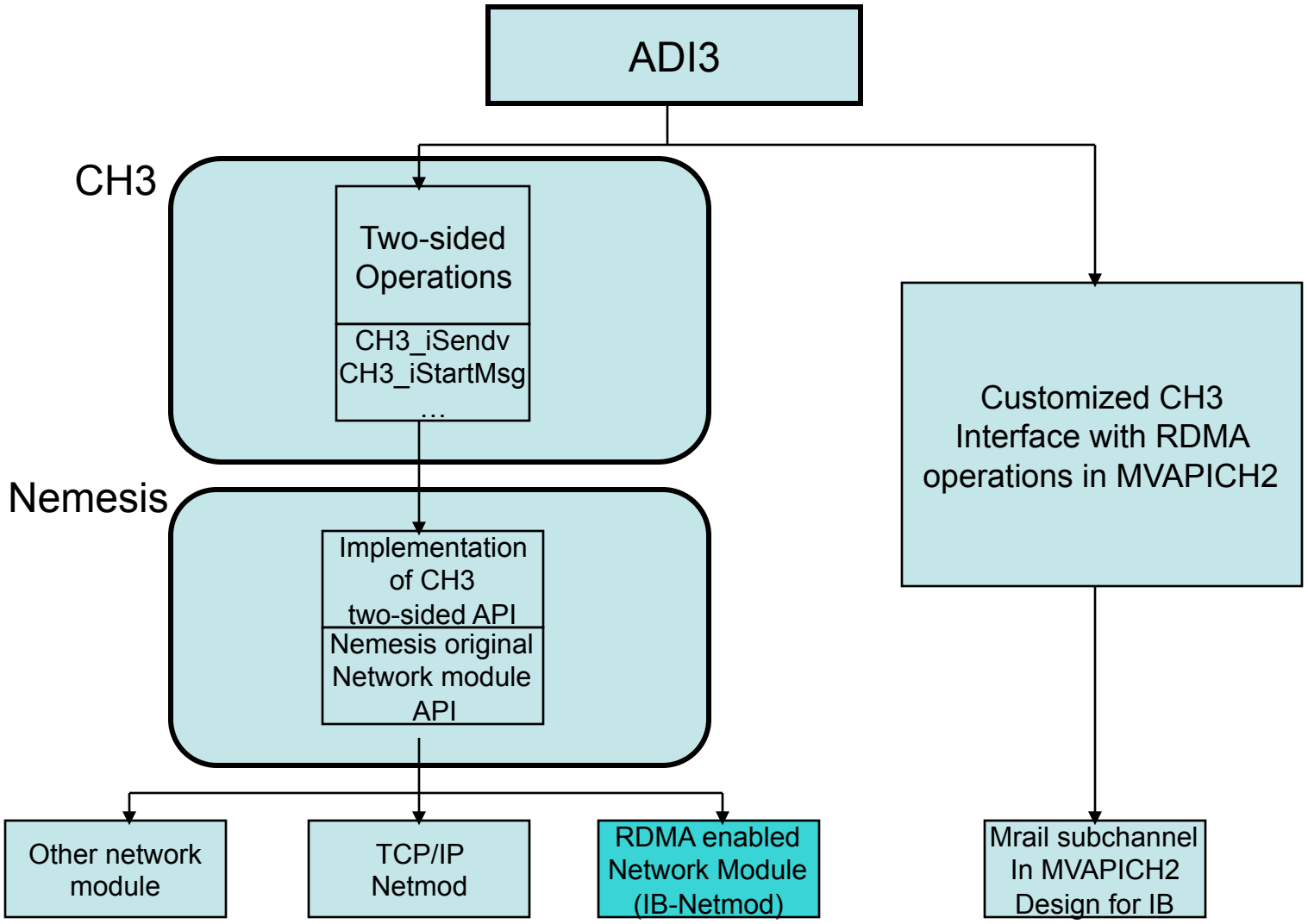
- Introduction
- Problem Statement
- Design Challenges
- Evaluation of Performance
- Conclusions and Future Work

Designing IB Support for Nemesis: IB-Netmod

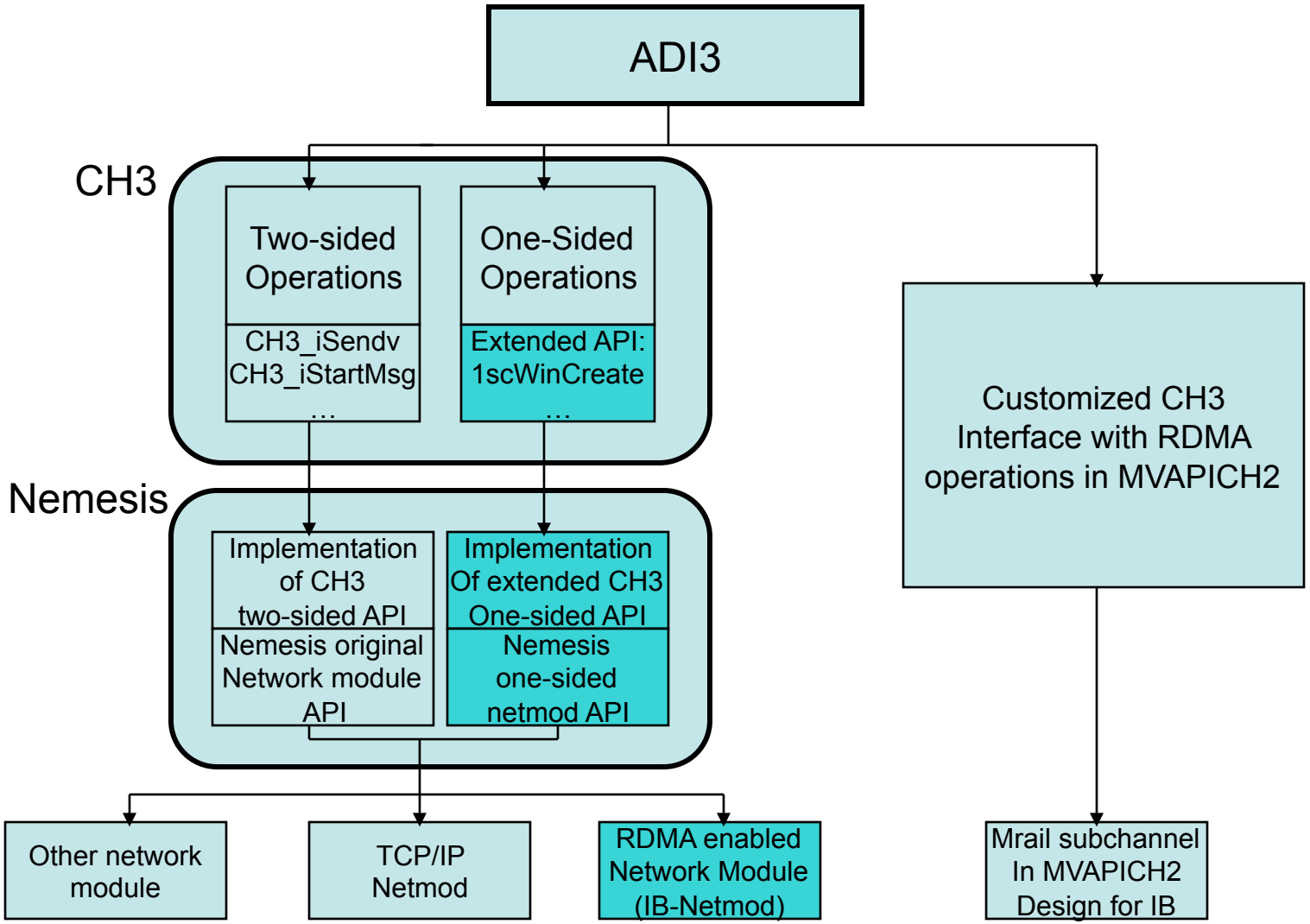
- Credit-based InfiniBand Netmod Header
- Additional Optimization Techniques.
  - SRQ
  - RDMA Fast Path
  - Header caching
- Limitation from existing API?
  - Stops directly one-sided supports from lower layer!



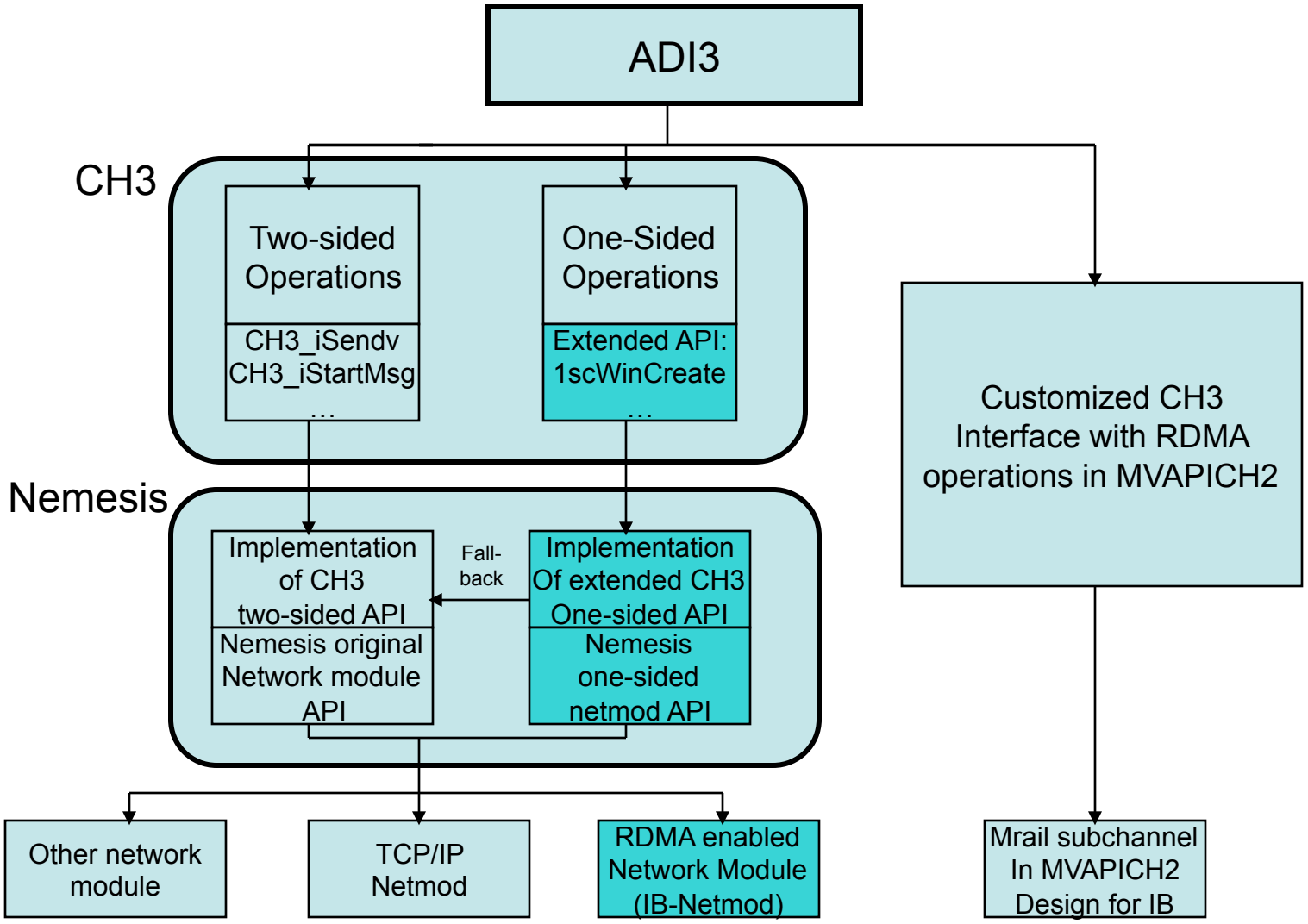
Proposed Extensions to Nemesis



Proposed Extensions to Nemesis



Proposed Extensions to Nemesis



Extended CH3 One-sided API

- CH3_1scWinCreate(void *base, MPI-Aint size, MPID_Win *win_ptr, MPID_Comm *comm_ptr):
  - Get window object handler and initial address of the window
- CH3_1scWinPost(MPID_Win *win_ptr, int *group);
  - Implement or be aware of the starting of a RMA epoch
- CH3_1scWinWait(MPID_Win *win_ptr)
  - Check the completion of an RMA epoch as a target.
- CH3_1scWinFinish(MPID_Win *win_ptr)
  - Inform remote processes about the finish of all RMA operations in current epoch.
- CH3_1scWinPut(MPID_Win *win_ptr, MPIDI_RMA_ops *rma_op)
  - Interface for sub-channels to realize truly one-sided put operations.
- CH3_1scWinGet(MPID_Win *win_ptr, MPIDI_RMA_ops *rma_op)

Extended Nemesis One-sided API

- MPID_nem_net_mod_WinCreate(void *base, MPI_Aint size, int comm_size, int rank, MPID_Win **win_ptr, MPID_Comm *comm_ptr)
  - Interface for netmods to get prepared for truly one-sided operations.
- MPID_nem_net_mod_WinPost(MPID_Win *win_ptr, int target_rank)
  - Interface for netmods with RMA ability to realize sync by RDMA write or even hardware multicast features.
- MPID_nem_net_mod_WinFinish(MPID_Win *win_ptr)
  - Interface for netmods with RDMA ability to realize CH3_1scWinFinish by RDMA write.
- MPID_nem_net_mod_WinWait(MPID_Win *win_ptr)
  - Interface for netmods to match net_mod_WinFinish functions with proper polling schemes.
- MPID_nem_net_mod_Put(MPID_Win *win_ptr, MPIDI_RMA_ops *rma_op, int size) MPID_nem_net_mod_Get(MPID_Win *win_ptr, MPIDI_RMA_ops *rma_op, int size)
  - Interface for netmods to carry out truly RMA put operation by hardware features.

Outline

- Introduction
- Problem Statement
- Design Challenges
- Evaluation of Performance
- Conclusions and Future Work

MVAPICH2 Software

- High Performance MPI Library for IB and 10GE
  - MVAPICH (MPI-1) and MVAPICH2 (MPI-2)
  - Used by more than 1,250 organizations
  - Empowering many TOP500 clusters
  - Available with software stacks of many IB, 10GE and server vendors including Open Fabrics Enterprise Distribution (OFED)
  - Also supports uDAPL device
  - http://mvapich.cse.ohio-state.edu
  - IB-Netmod has been incorporated into MVAPICH2 since 1.5 release (July 2010); IB-Netmod with one-sided extension will be available in the near future.

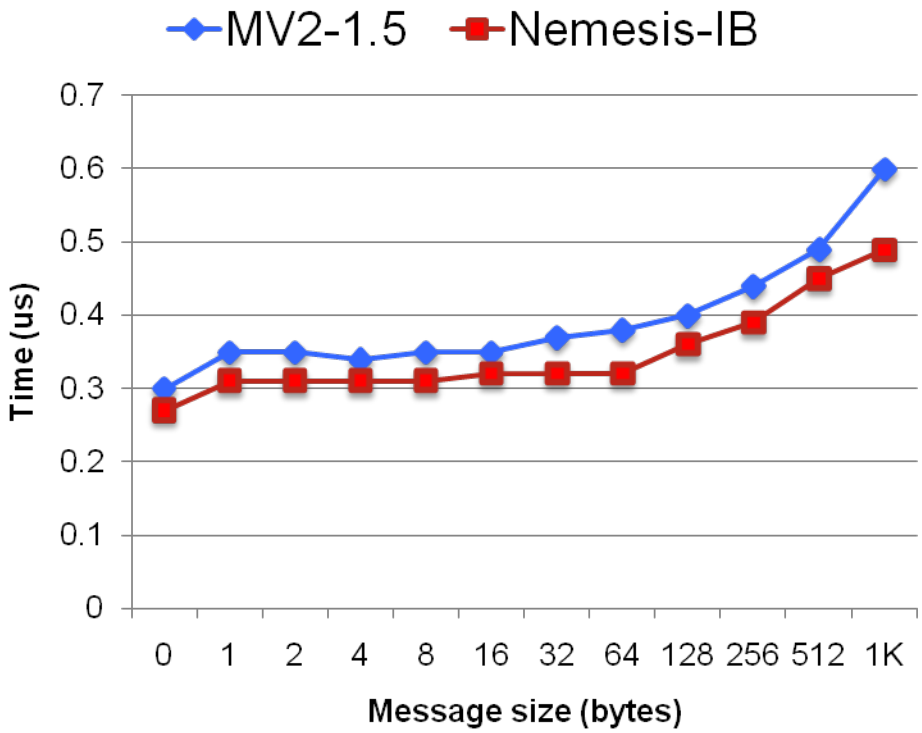
Experimental Testbed

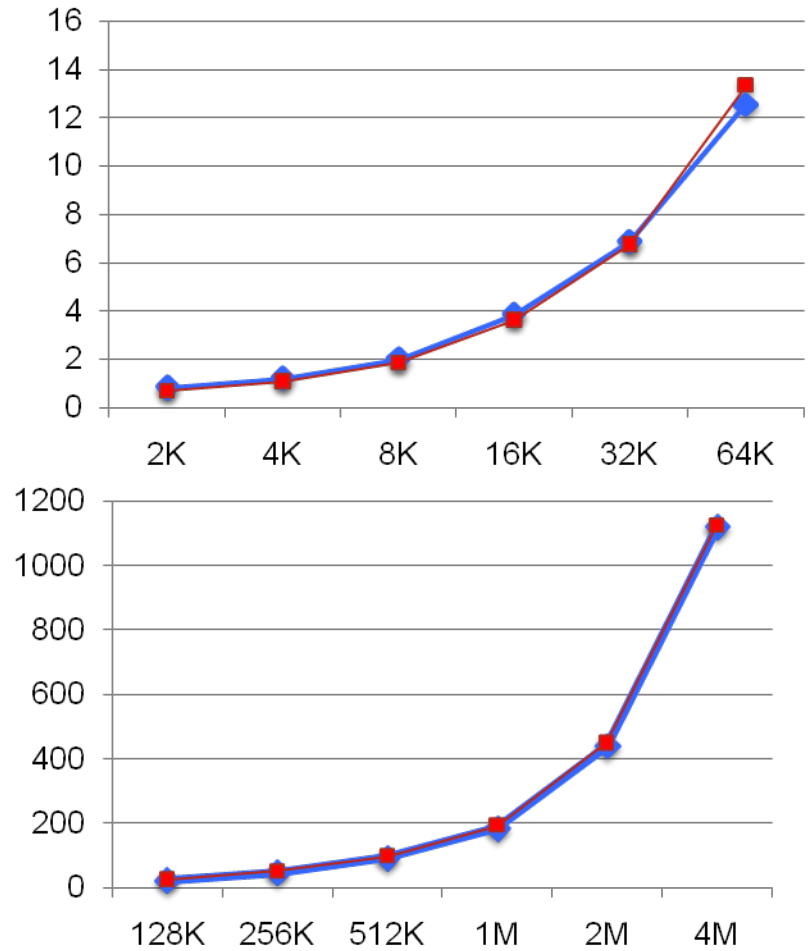
- Cluster A:
  - 8 Intel Nehalem machines
  - ConnectX QDR HCAs
  - Eight Intel Xeon 5500 processors
  - two sockets of four cores
  - 2.40 GHz with 12 GB of main memory.
- Cluster B:
  - 32 Intel Clovertown
  - ConnectX DDR HCAs
  - Eight Intel Xeon processors
  - 2.33 GHz with 6 GB of main memory.
- RedHat Enterprise Linux Server 5, OFED version 1.4.2.

Results Evaluation

- Micro-benchmark Level Evaluation
  - Two-sided
  - One-sided
  - Available Overlap rate
- Application Level Evaluation
  - NAMD
  - AWP-ODC

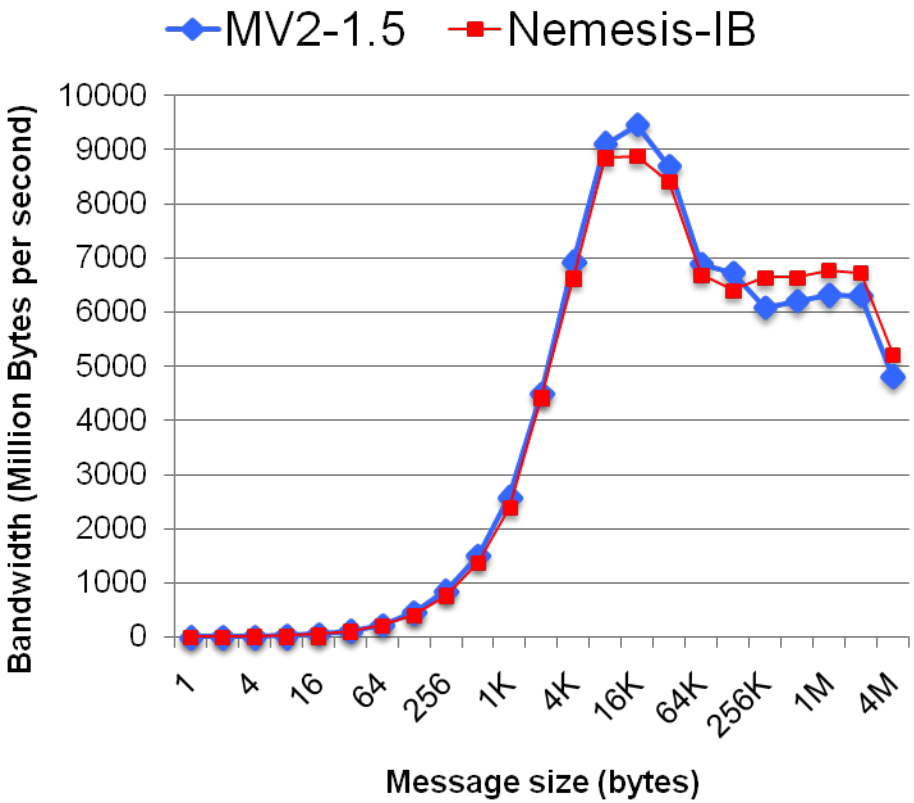
Micro-benchmark Evaluation Two-sided Intra-node Latency

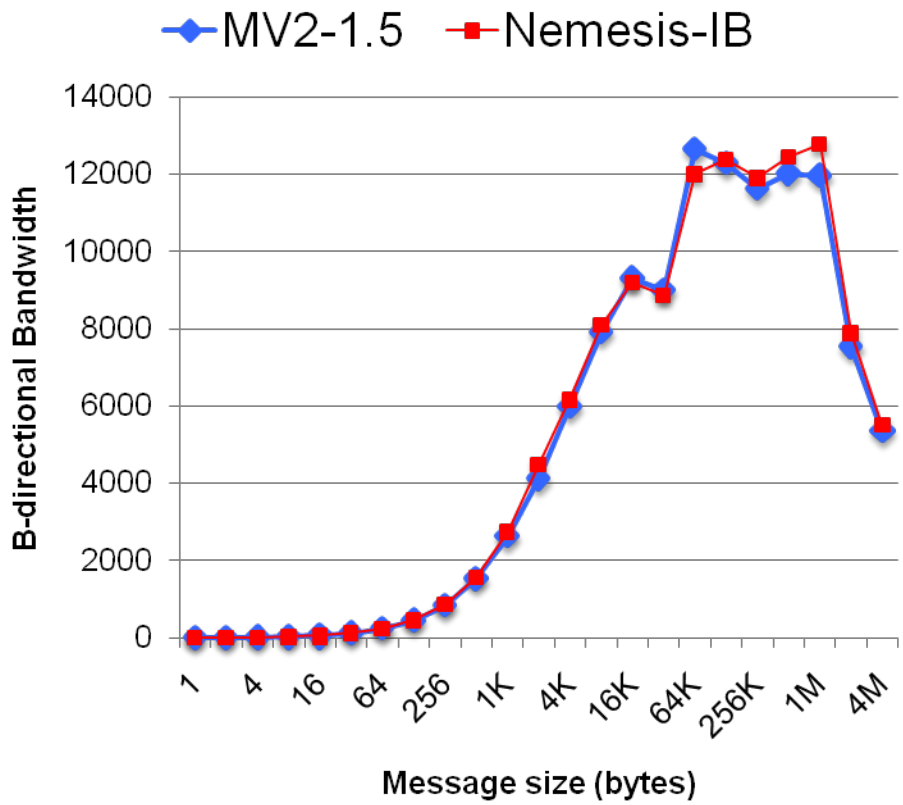




- Nemesis intra-node communication design helps to reduce the latency of small messages.

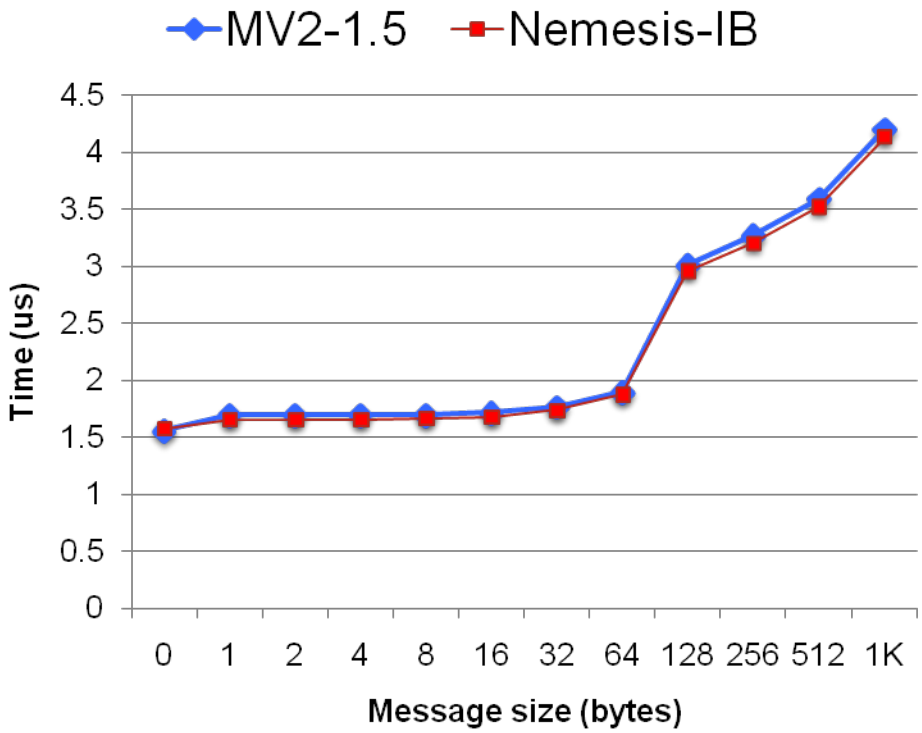
Micro-benchmark Evaluation
Two-sided Intra-node Bandwidth

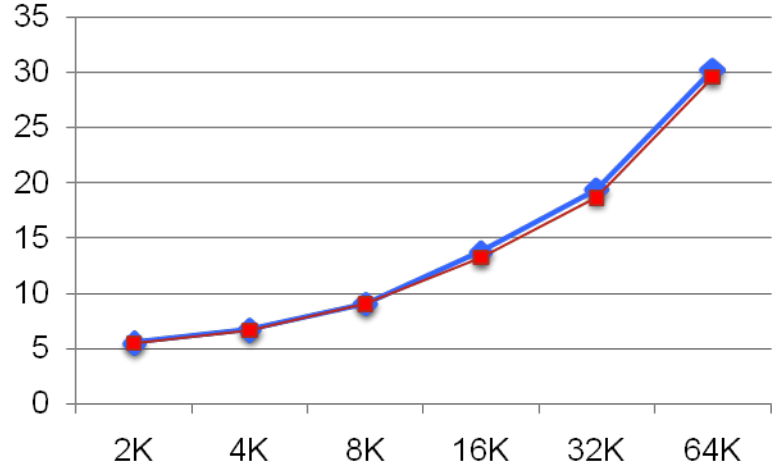


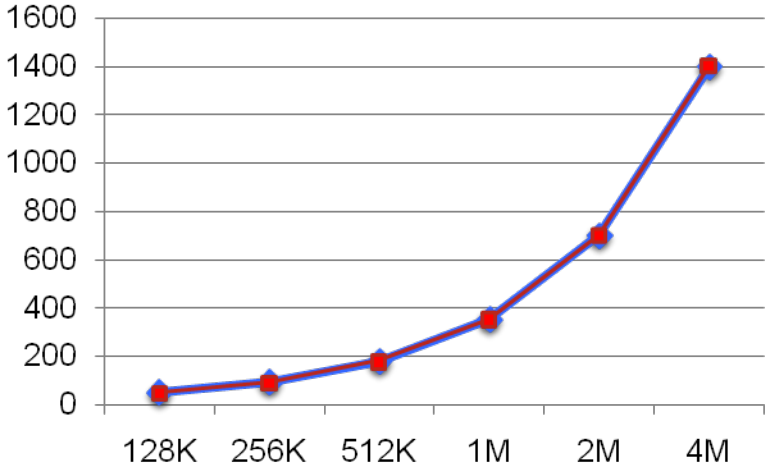


- Between 8KB and 128KB message size range, MVPICH2 1.5 with LiMIC2 performs better.
- For even larger messages, Nemesis with KNEM has average 400MB/s larger bandwidth.
- Different inner design of KNEM and LiMIC2.

Micro-benchmark Evaluation Two-sided Inter-node Latency

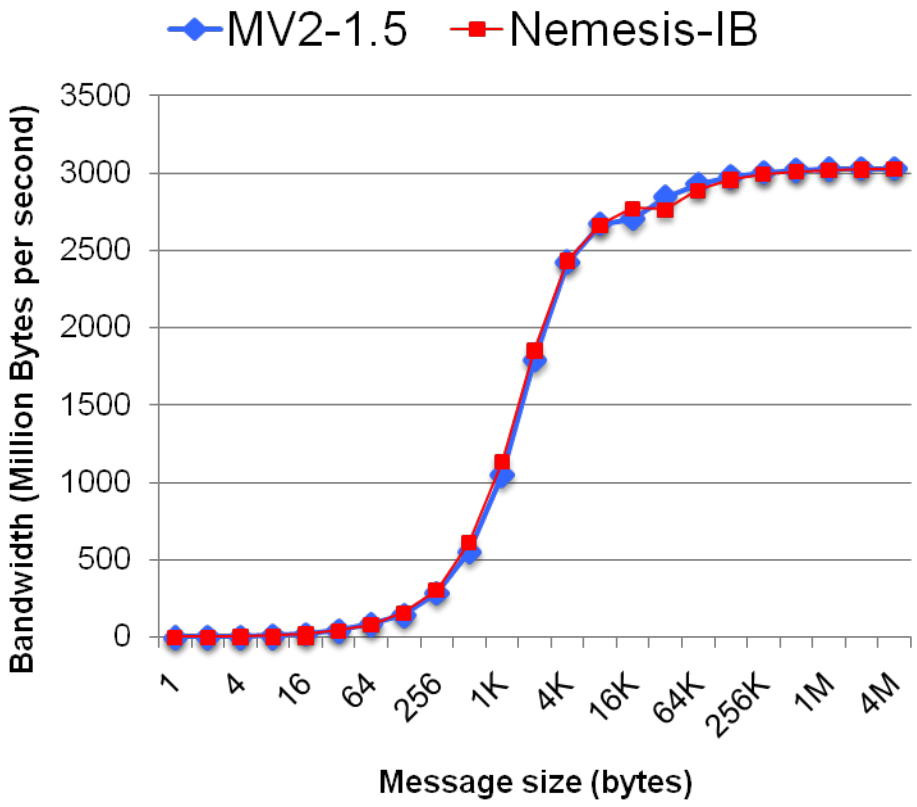


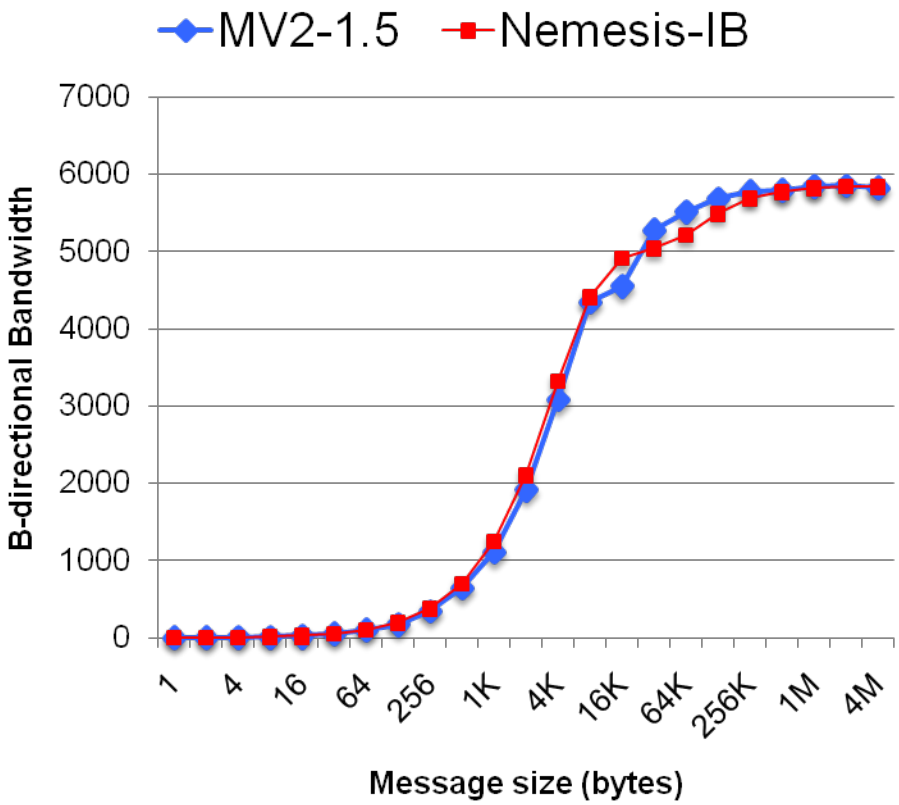




- IB-netmod is able to provide 1.5us latency by using native InfiniBand, which efficiently utilize the high performance of InfiniBand network.
- Comparable performance as MVAPICH2 1.5

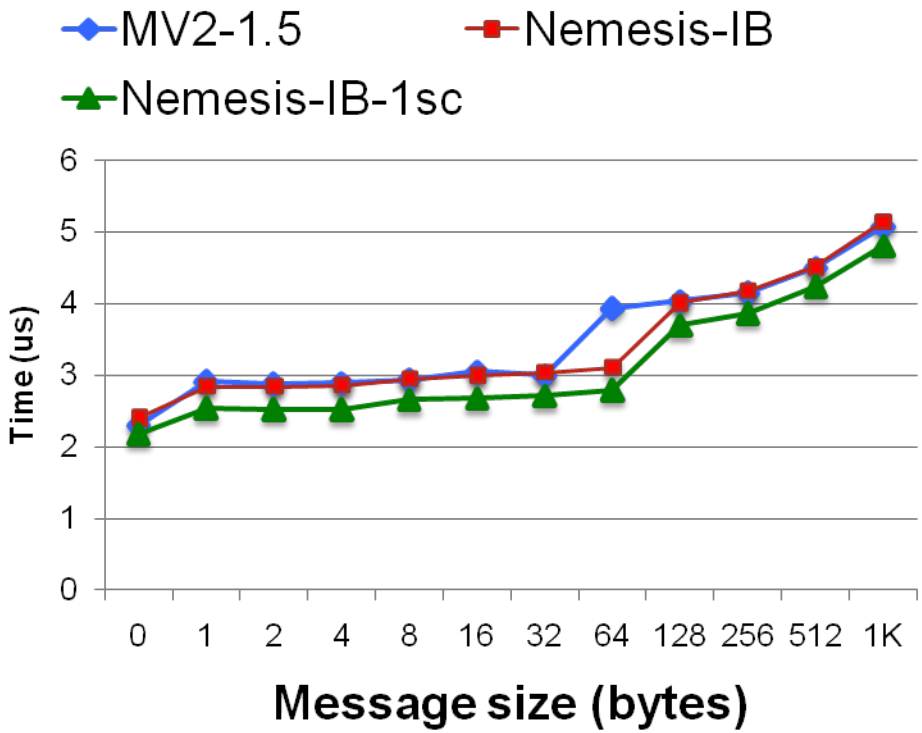
Micro-benchmark Evaluation
Two-sided Inter-node Bandwidth

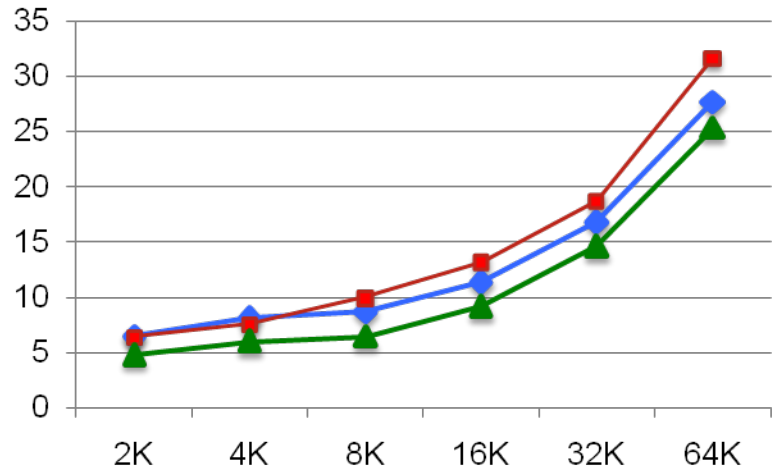


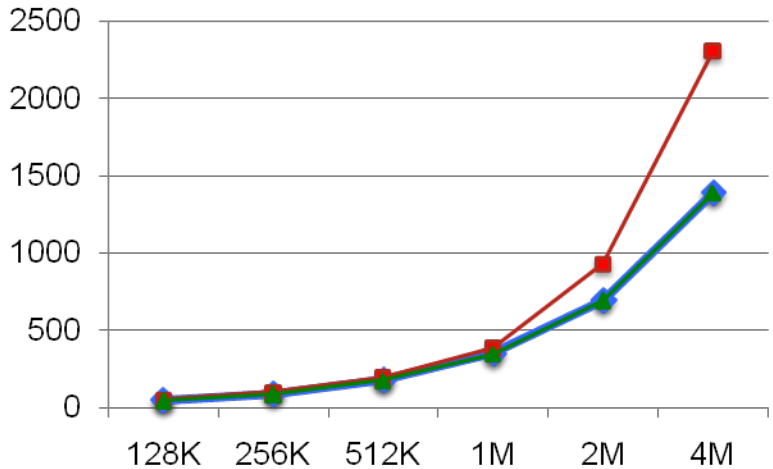


- Though IB-Netmod can achieve even better bi-directional bandwidth for medium message sizes up to 16K Bytes, it loses up to 200MB/s performance for message range between 32K Bytes and 256K Bytes.

Micro-benchmark Evaluation One-Sided MPI_Put Latency

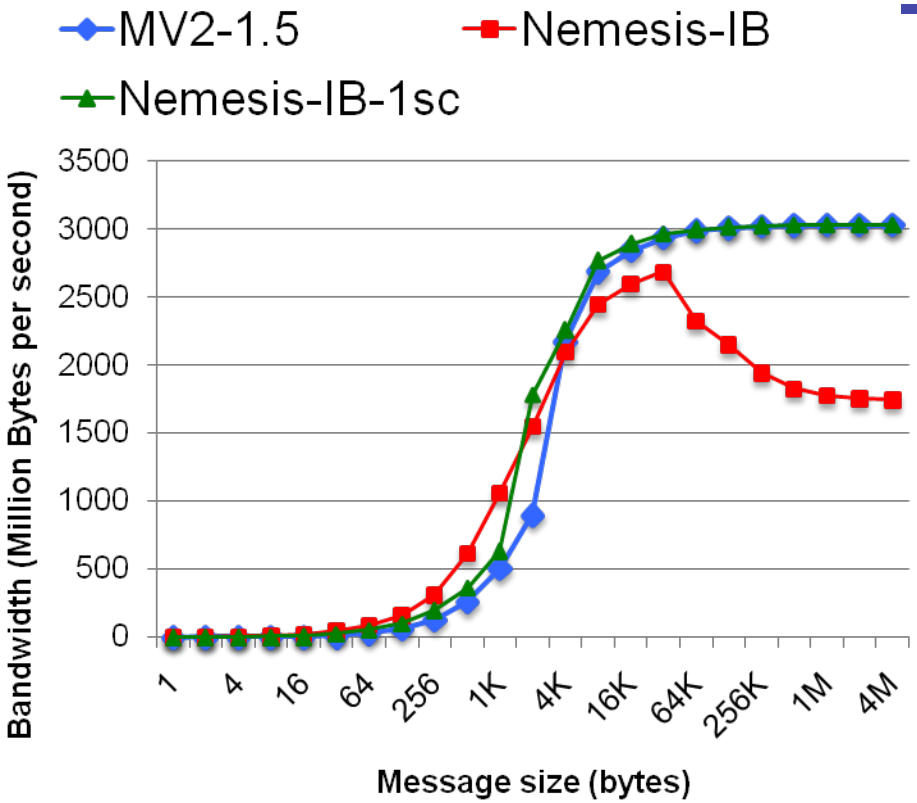


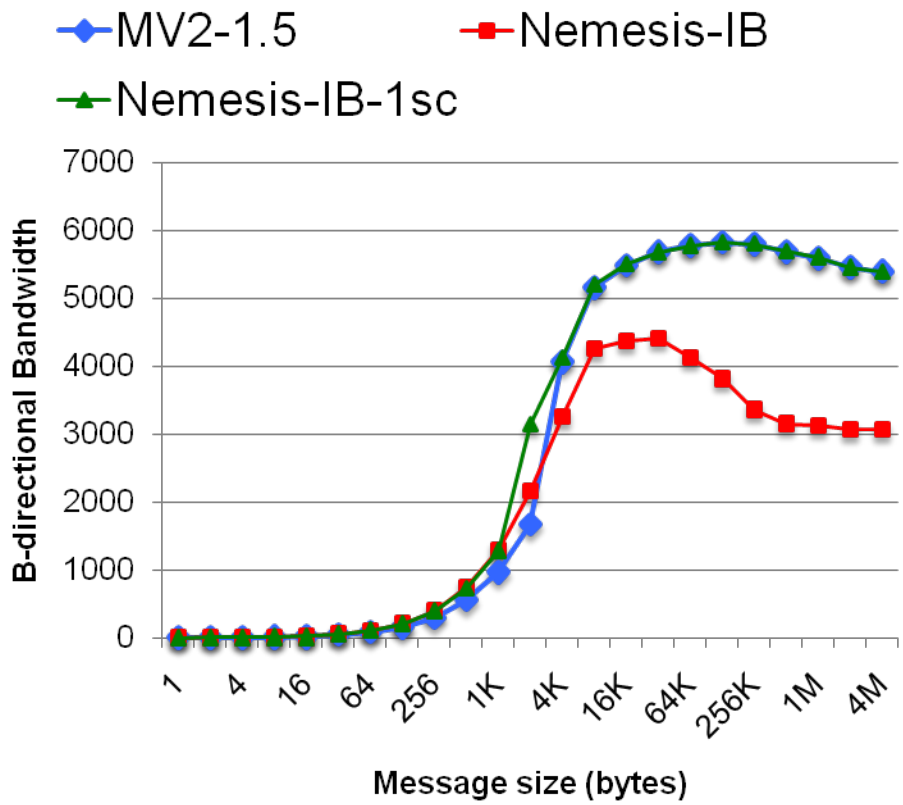




- Through extended API, Nemesis IB-Netmod is able to reduces an average 10% latency for small messages.
- Extended API eliminates the fall-back overhead of customized CH3 interfaces..

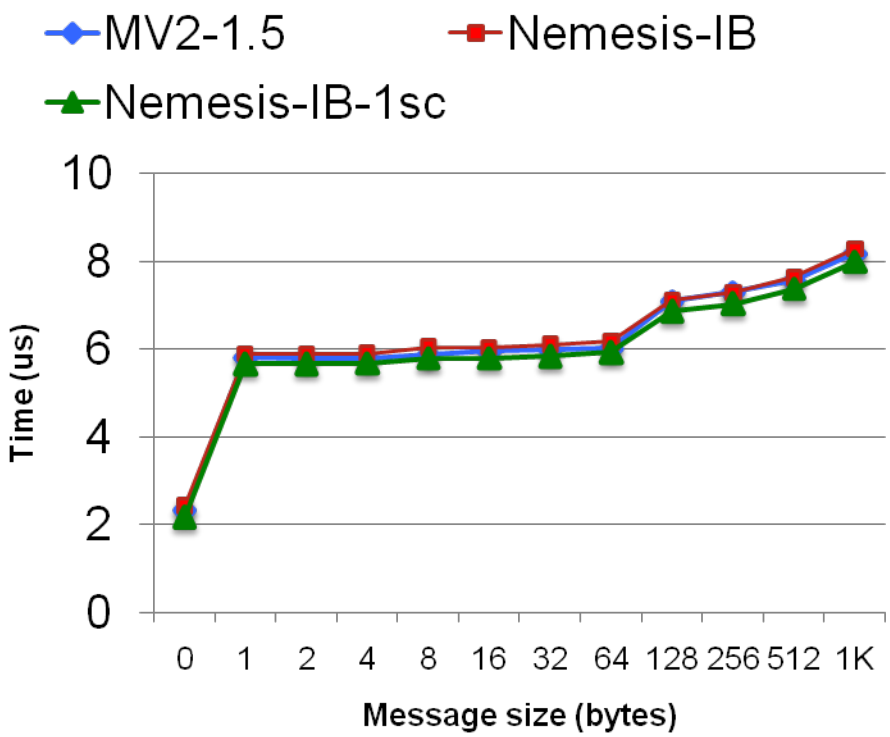
Micro-benchmark Evaluation One-Sided MPI_Put Bandwidth

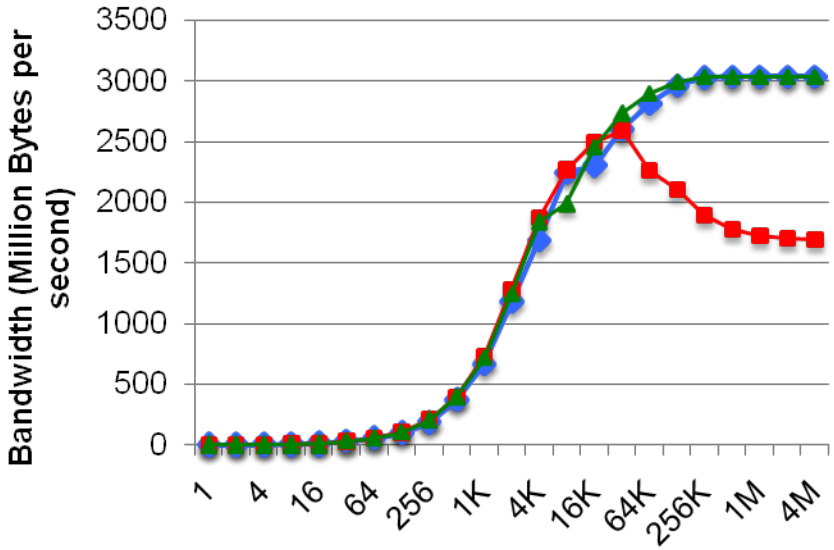


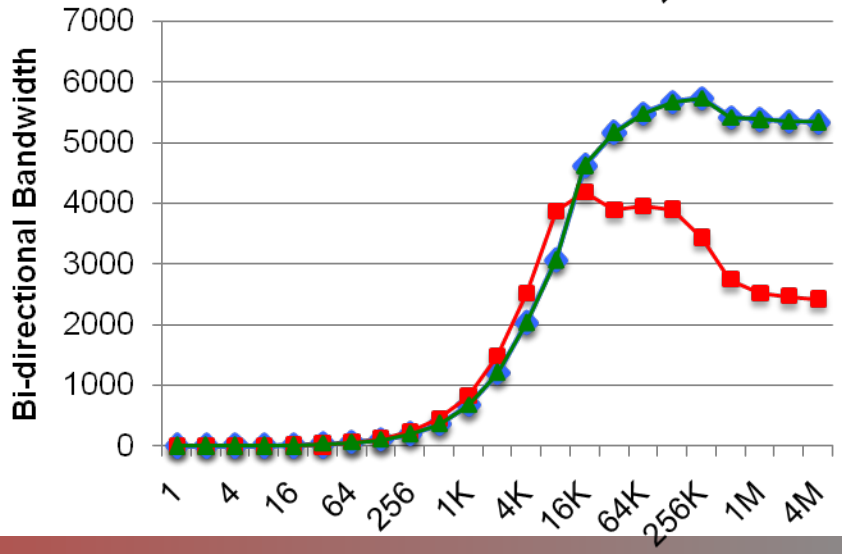


- By direct one-sided implementation of MPI_Put operation, Nemesis-IB with extended one-sided API achieve nearly full bandwidth, the same as MVAPICH2 1.5.
- Nemesis IB-Netmod with original two-sided based API can only achieve 60% of full bandwidth.

Micro-benchmark Evaluation One-Sided MPI_Get

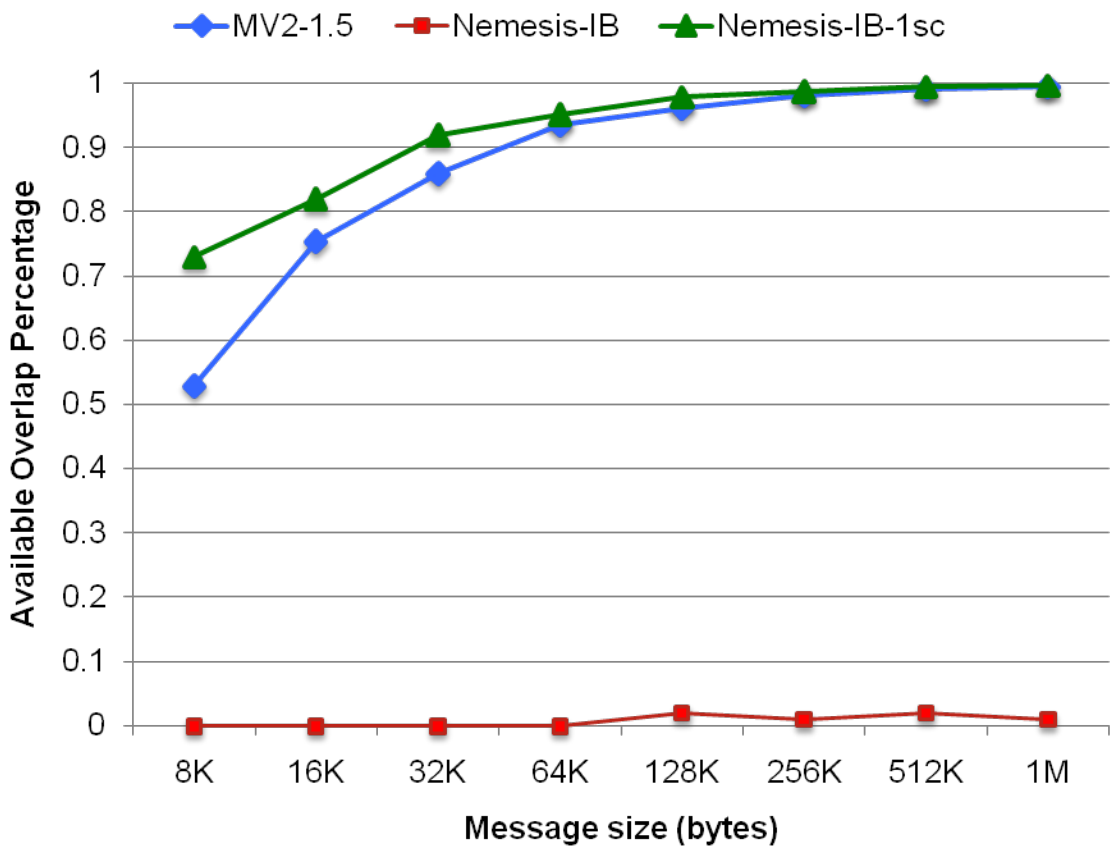






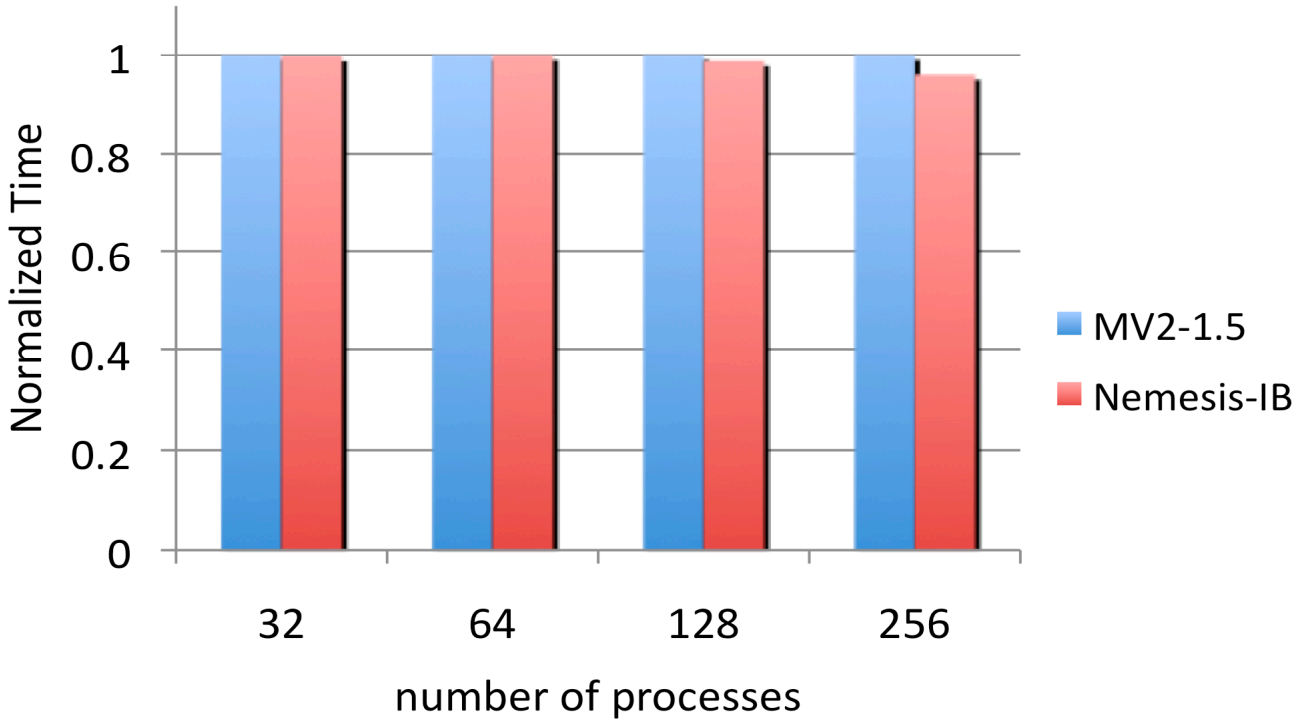
- Similar results in MPI_Get benchmark.

Micro-benchmark Evaluation



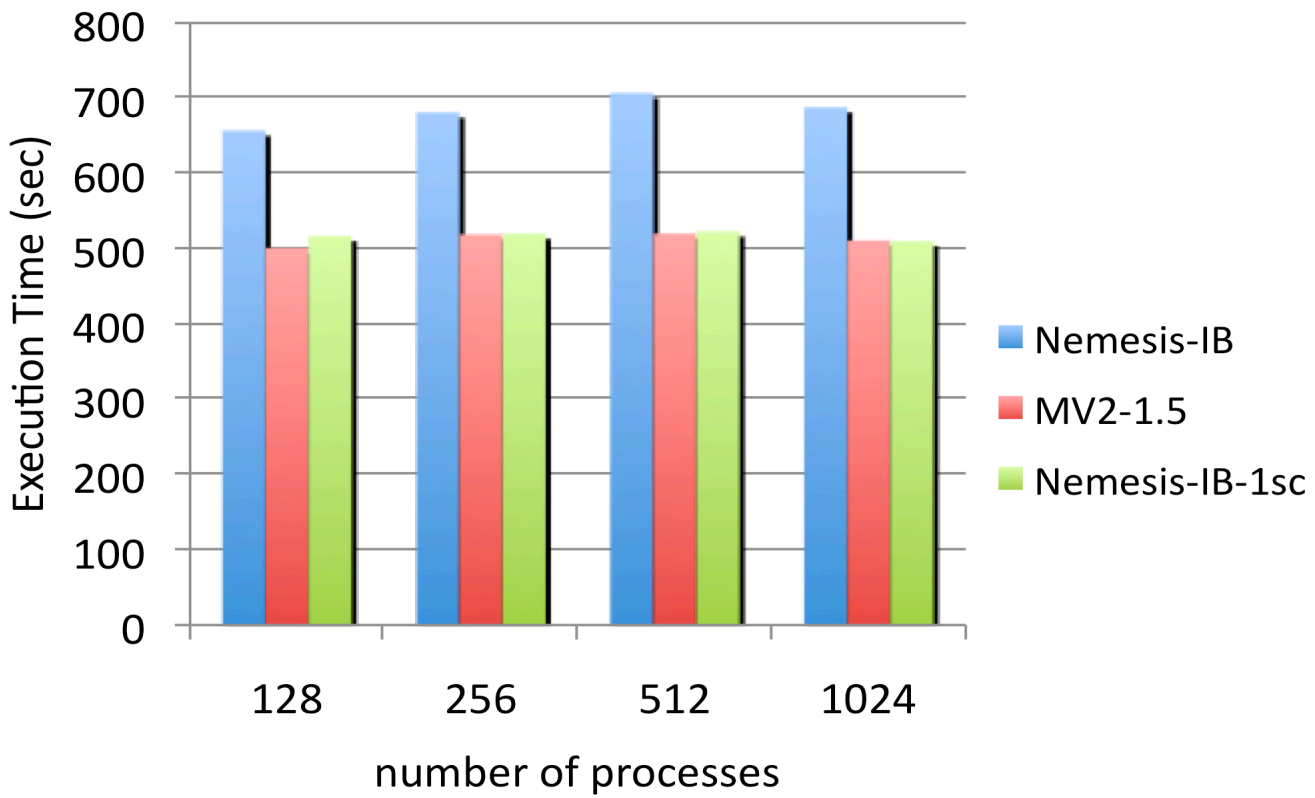
- Computation is inserted after each round of multiple Put or Get operations.
- Overlap = (Tcomm + Tcomp - Ttotal)/Tcomm
- 90% overlap achieved for large message, through extended API.

Application Evaluation NAMD apoa1



- Production molecular dynamics program for high performance simulation of large bio-molecular system.
- Nemesis IB-Netmod performs as much good as MVAPICH2 1.5.
- As the number of processes increase, the new IB-Netmod shows a trend of even better performance, which maybe due to Nemesis intra-node optimization.

Application Evaluation AWP-ODC



- Anelastic Wave Propagation: earthquake simulation application.
- http://hpgeoc.sdsc.edu/AWPODC/
- AWP-ODC one-sided version with 128*256*256 elements per process.
- 24% reduction of execution time.

Conclusion

- **InfiniBand** based network module
  - based on **MVAPICH2**
  - for modular **Nemesis** communication layer
- **Extended** Nemesis API
  - **truly** one-sided communication support for RMA semantics.
  - **Implemented** in the new Nemesis IB-Netmod.
  - **Evaluation** of its impact comparing with MVAPICH2 1.5.
- **Reusability**?
  - We believe the extended API can also be utilized by other netmods.

Future Work

- **Intra-node** one-sided communications
- IB-Netmod:
  - **Scalability**
  - **Performance** optimization techniques.
- Continue to design and evaluate new interfaces.

Thanks!

{luom, potluri, laipi, mancini, subramon, kandalla, surs, panda}@cse.ohio-state.edu

MVAPICH

Network-based Computing Laboratory

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