Unified Runtime for PGAS and MPI over OFED

D. K. Panda and Sayantan Sur

Network-Based Computing Laboratory

Department of Computer Science and Engineering

The Ohio State University, USA

OpenFabrics Monterey Workshop (April '11)

Outline

- Introduction
- Challenges in unifying UPC and MPI
- Design Solutions
- Experimental Results & Analysis
- Summary

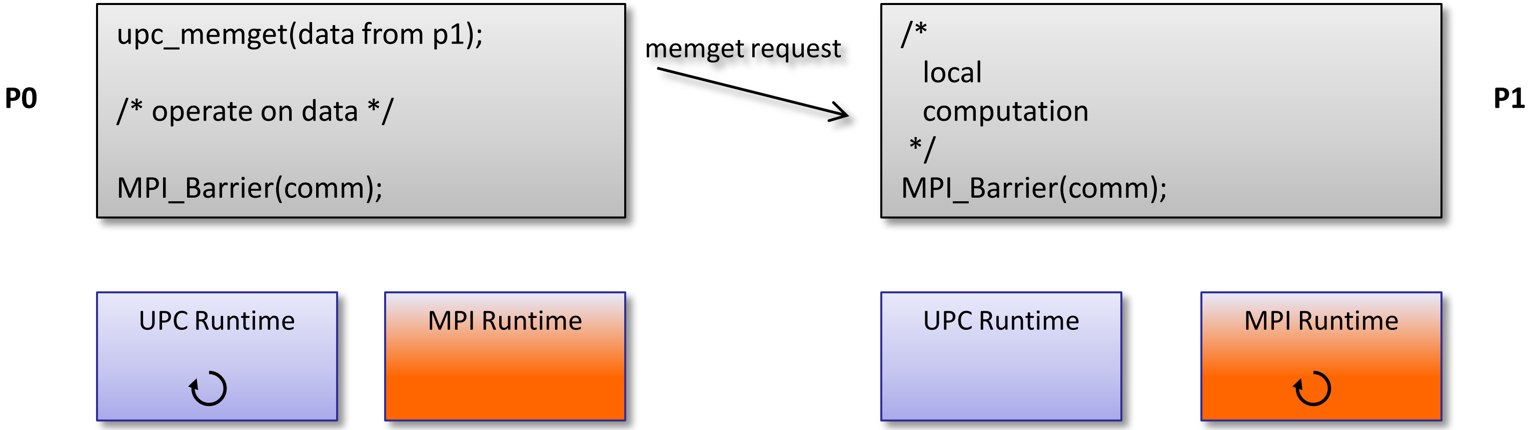
Introduction

- Partitioned Global Address Space (PGAS) programming model is gaining interest
  - Global view improves programmer productivity
  - Language and Compiler support improves performance
- Unified Parallel C (UPC) is one popular PGAS language
  - Scientists and developers want to use it over OFED

Introduction (Cont’d)

- There are several problems:
  - Parts of big applications and third party libraries use MPI
  - Parallel Math and Physics libraries have very high investment, ***cannot*** re-write them!
  - MPI and UPC currently don’t interoperate very well
  - Issues with performance and scalability of UPC runtime on OFED
  - No unified runtime to support both MPI and UPC over OFED with best performance and scalability
    - Current performance comparison between MPI and UPC is misleading
  - No unified runtime to design hybrid programs (MPI and UPC) on emerging multi-core environments

Why doesn't UPC work with MPI?



P0

```
upc_memget(data from p1);

/* operate on data */

MPI_Barrier(comm);
```

```
/*
  local
  computation
 */
MPI_Barrier(comm);
```

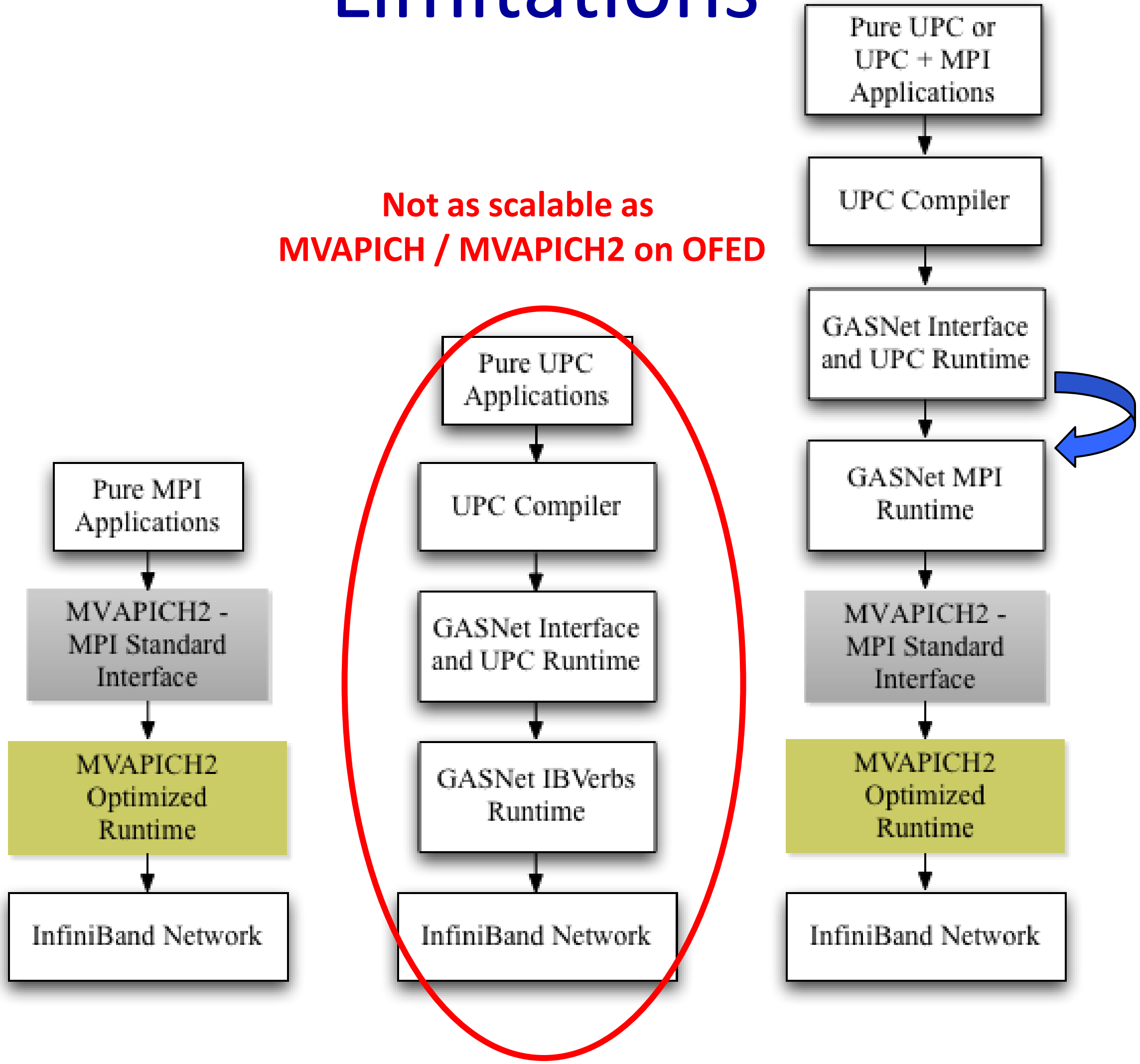
P1

- Deadlock: message in one runtime, but application waits in other runtime
- Current prescription to avoid this is to *barrier* in one mode (either UPC or MPI) before entering the other
- Bad performance!!

Outline

- Introduction
- Challenges in unifying UPC and MPI
- Design Solutions
- Experimental Results & Analysis
- Summary

Various ways to use UPC and MPI and Limitations


Pure MPI Applications
MVAPICH2 - MPI Standard Interface
MVAPICH2 Optimized Runtime
InfiniBand Network
Pure UPC Applications
UPC Compiler
GASNet Interface and UPC Runtime
GASNet IBVerbs Runtime
InfiniBand Network
Pure UPC or UPC + MPI Applications
UPC Compiler
GASNet Interface and UPC Runtime
GASNet MPI Runtime
MVAPICH2 - MPI Standard Interface
MVAPICH2 Optimized Runtime
InfiniBand Network

Not as scalable as MVAPICH / MVAPICH2 on OFED

UPC semantics Mapped onto MPI

Bad performance!!

What is the way forward?

- Can we place UPC on top of MPI?
  - Active messages (AM) not part of MPI; critical to UPC
  - UPC is lighter-weight, so putting on top of MPI loses performance
  - Other model mismatches (some may be solved by MPI-3)

- *Path forward: unify runtimes, not programming models*

Problem Statement

- Can we design a communication library for UPC?
  - Scalable on large InfiniBand clusters
  - Provides equal or better performance than existing runtime
- Can this library support both MPI and UPC?
  - Individually, both with great performance
  - Simultaneously, with great performance and less memory

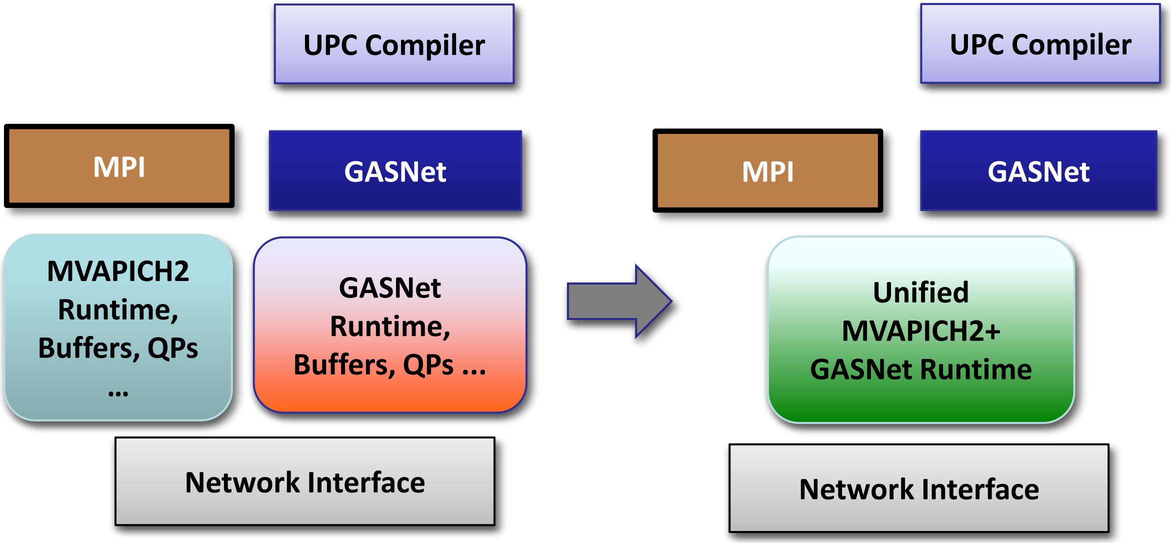
Benefits

- Allow scientists to develop applications in the following modes
  - MPI only
  - PGAS (UPC) only
  - Hybrid (MPI and UPC)
- Allow scientists to evaluate the impact of programming models on applications on next generation systems in a fair manner

Outline

- Introduction
- Challenges in unifying UPC and MPI
- Design Solutions
- Experimental Results & Analysis
- Summary

Overall Approach

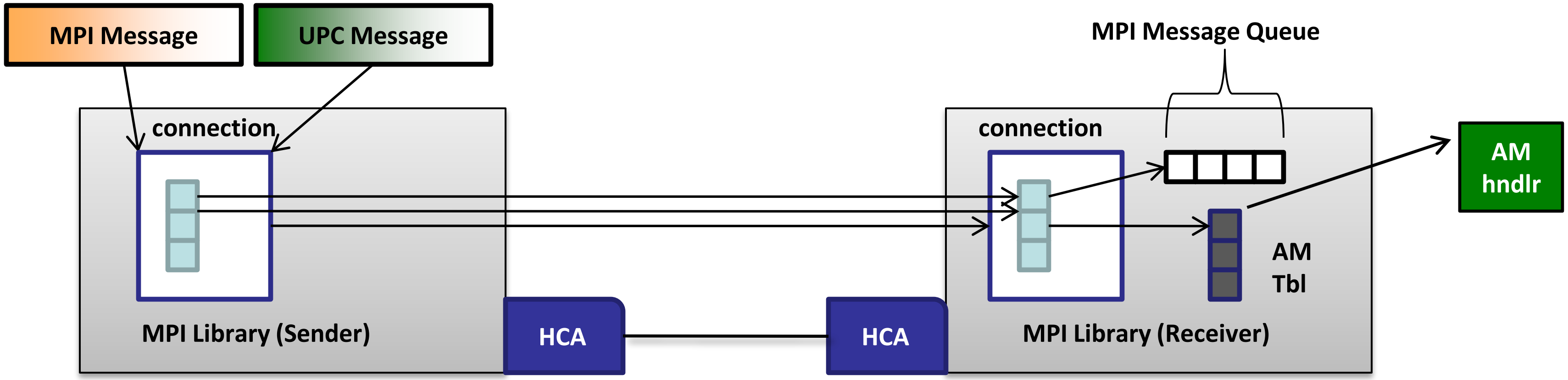

UPC Compiler
MPI
GASNet
MVAPICH2 Runtime, Buffers, QPs …
GASNet Runtime, Buffers, QPs …
Network Interface
UPC Compiler
MPI
GASNet
Unified MVAPICH2+ GASNet Runtime
Network Interface

- Unified runtime provides APIs for MPI and GASNet
- **UCR** (Unified Communication Runtime)

The UCR Interface

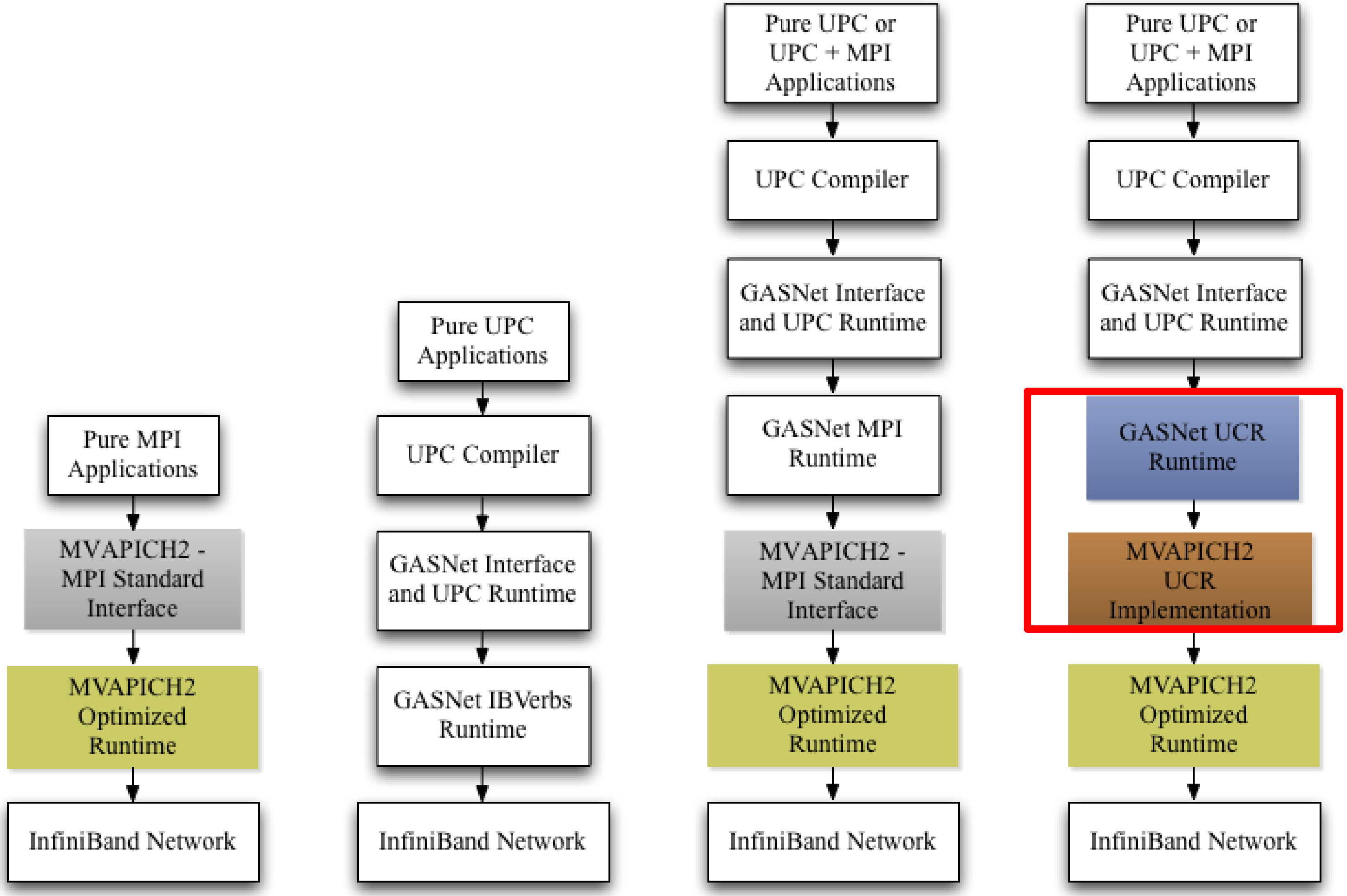
- Different Active Message (AM) APIs based on size for optimization
  - Send short AM without arguments
  - Short AM (no data payload)
  - Medium AM (bounce buffer using RDMA Fast Path)
  - Large AM (RDMA Put, on-demand connections)
- GASNet Extended interface for efficient Remote Memory Access (RMA)
  - Inline put
  - Put (may be internally buffered)
  - Put bulk (send buffer will not be touched, no buffering)
  - Get (RDMA Read)

Unified Implementation



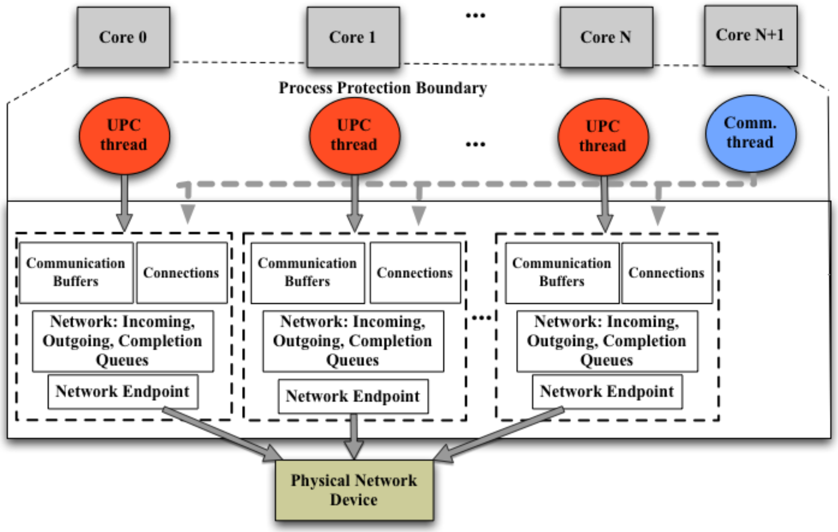
- All resources are shared between MVAPICH2 and UPC
  - Connections, buffers, memory registrations
  - Schemes for establishing connections (fixed, on-demand)
  - RDMA for large AMs and for PUT, GET

New Configuration for UPC and MPI

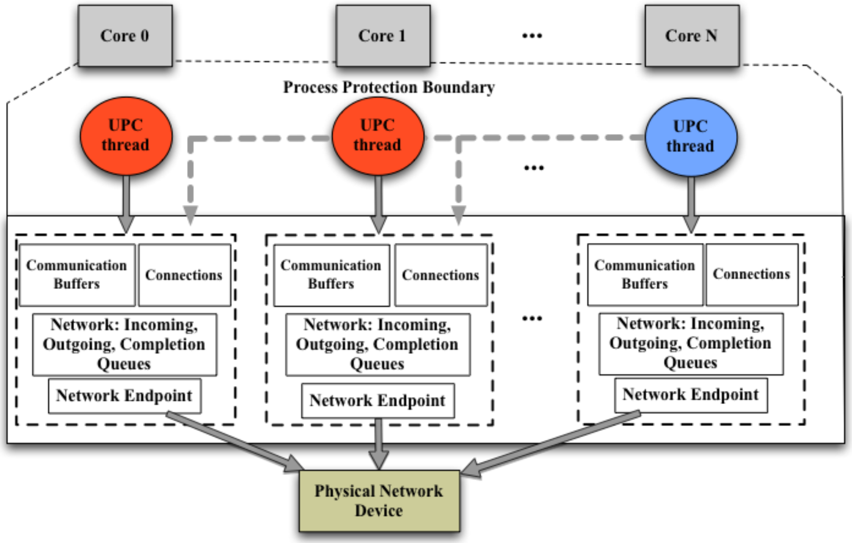

Pure MPI Applications
MVAPICH2 - MPI Standard Interface
MVAPICH2 Optimized Runtime
InfiniBand Network
Pure UPC Applications
UPC Compiler
GASNet Interface and UPC Runtime
GASNet IBVerbs Runtime
InfiniBand Network
Pure UPC or UPC + MPI Applications
UPC Compiler
GASNet Interface and UPC Runtime
GASNet MPI Runtime
MVAPICH2 - MPI Standard Interface
MVAPICH2 Optimized Runtime
InfiniBand Network
Pure UPC or UPC + MPI Applications
UPC Compiler
GASNet Interface and UPC Runtime
GASNet UCR Runtime
MVAPICH2 UCR Implementation
MVAPICH2 Optimized Runtime
InfiniBand Network

Our Design

Lock Free Multi-threaded Runtime with Multiple Endpoints



Multi-threaded (Multi-endpoint) Runtime with communication thread



Multi-threaded (Multi-endpoint) Runtime with work stealing for load balancing

- Multi-network endpoint capable runtime
  - No network endpoint contention
  - Same performance as process based runtime
- Enables two new optimizations: Dedicated communication thread; work-stealing can now be enabled

Outline

- Introduction
- Challenges in unifying UPC and MPI
- Design Solutions
- Experimental Results & Analysis
- Summary

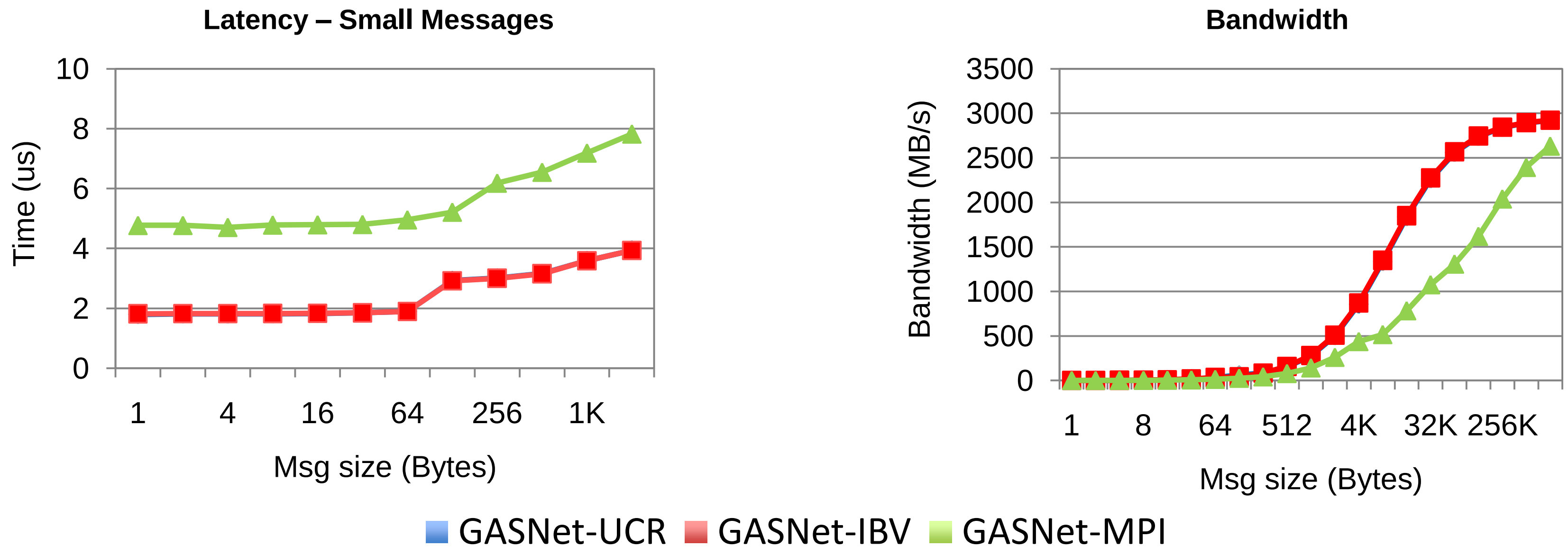
MVAPICH/MVAPICH2 Software

- High Performance MPI Library for IB, 10GE/iWARP & RoCE
  - MVAPICH (MPI-1) and MVAPICH2 (MPI-2)
  - Latest Releases: MVAPICH 1.2 and MVAPICH2 1.6
  - Used by more than 1,500 organizations in 60 countries
    - Registered at the OSU site voluntarily
  - More than 59,000 downloads from OSU site directly
  - Empowering many TOP500 production clusters during the last eight years
  - Available with software stacks of many IB, 10GE and server vendors including Open Fabrics Enterprise Distribution (OFED) and Linux Distros
  - Also supports uDAPL device to work with any network supporting uDAPL
  - http://mvapich.cse.ohio-state.edu/
- New design has been incorporated into MVAPICH2

Experimental Setup

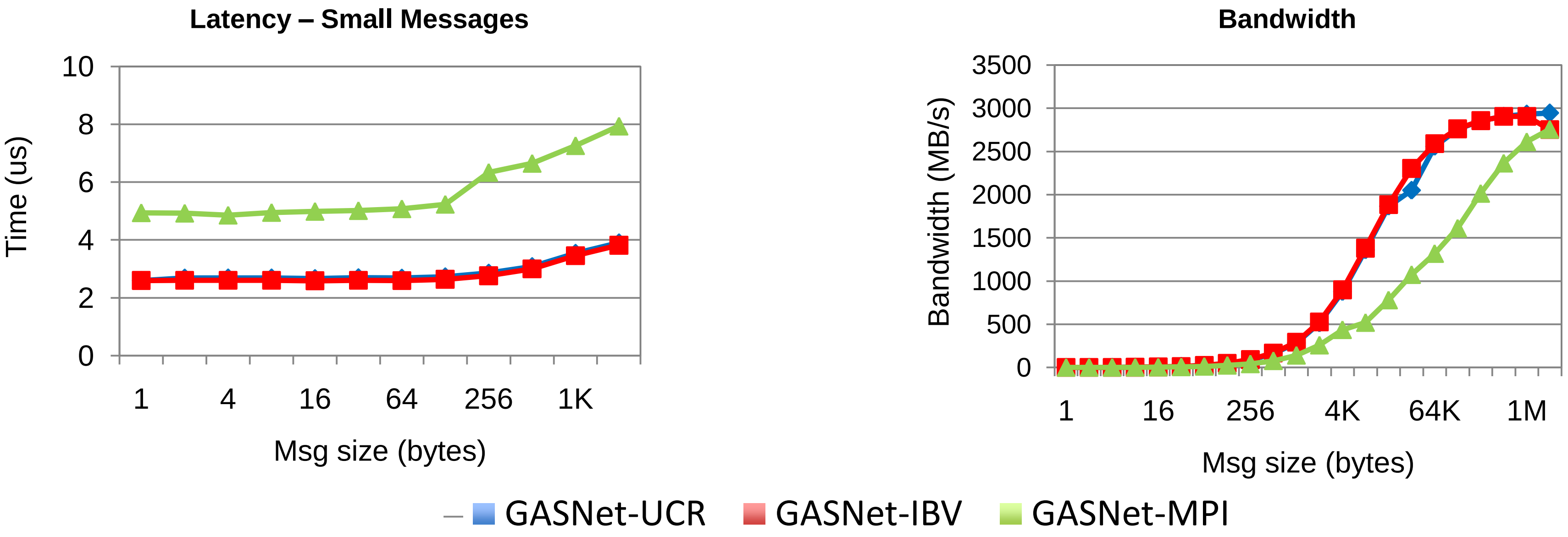
- Berkeley GASNet version 2.10.2 (--enable-pshm)
- Experimental Testbed
  - Type 1
    - Intel Nehalem (dual socket quad core Xeon 5500 2.4GHz)
    - ConnectX QDR InfiniBand
  - Type 2
    - Intel Clovertown (dual socket quad core Xeon 2.33GHz)
    - ConnectX DDR InfiniBand
  - Type 3
    - AMD Barcelona
    - Quad-socket quad-core Opteron 8530 processors
    - ConnectX DDR InfiniBand

Microbenchmark: upc_memput



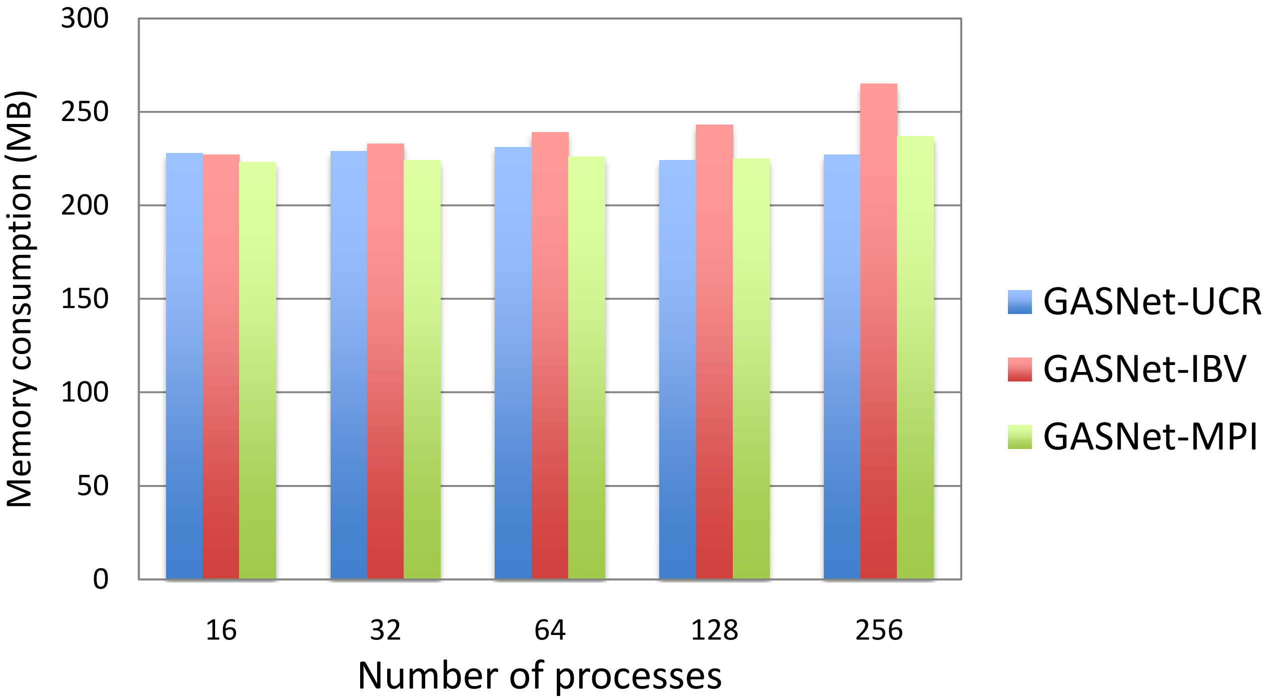
- Cluster #1 used for these experiments
- GASNet-UCR performs identically with GASNet-IBV
- Comparatively GASNet-MPI (i.e. UPC on top of MPI) performs worse
- Mismatch of Active Message semantics
    - Message queue processing overheads

Microbenchmark: upc_memget



- GASNet-UCR performs identically with GASNet-IBV
- Due to mismatch of AM semantics with MPI leads to worse performance

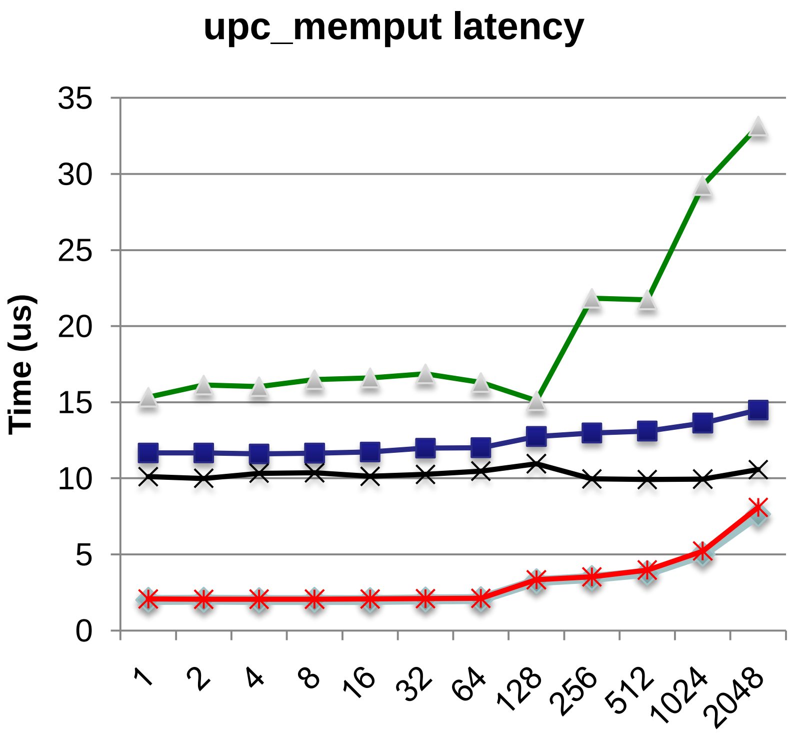
Memory Scalability

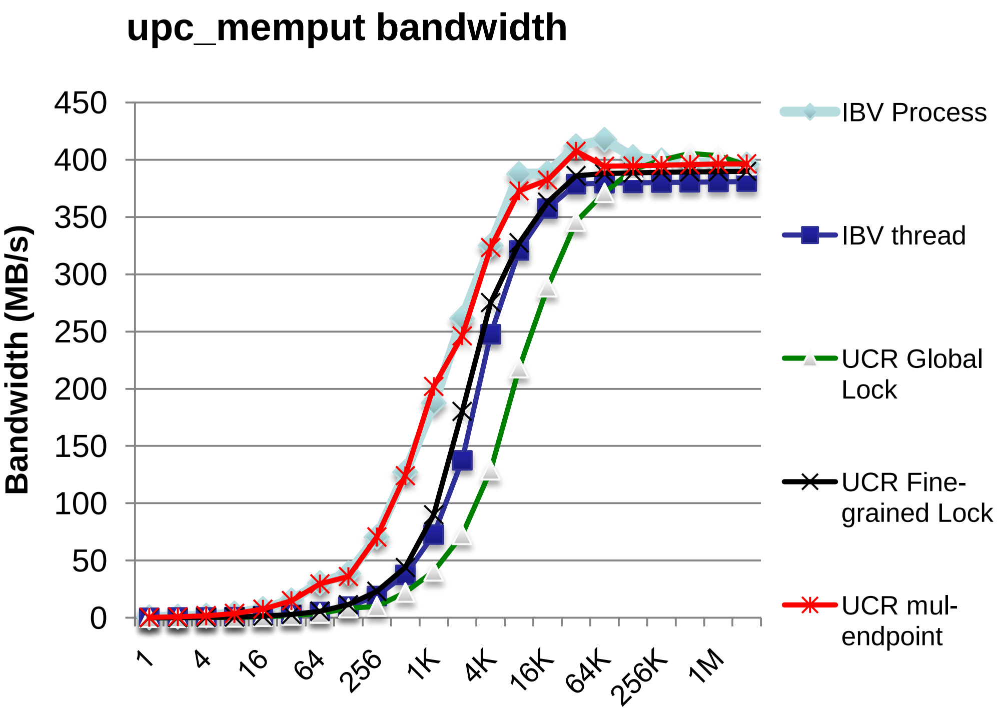


- UPC “hello world” program
- GASNet-IBV establishes all-to-all reliable connections
    - Not scalable (may be improved in future release)
- GASNet-UCR best scalability due to inherent hybrid design

J. Jose, M. Luo, S. Sur and D. K. Panda *“Unifying UPC and MPI Runtimes: Experience with MVAPICH”*, PGAS 2010, New York, New York

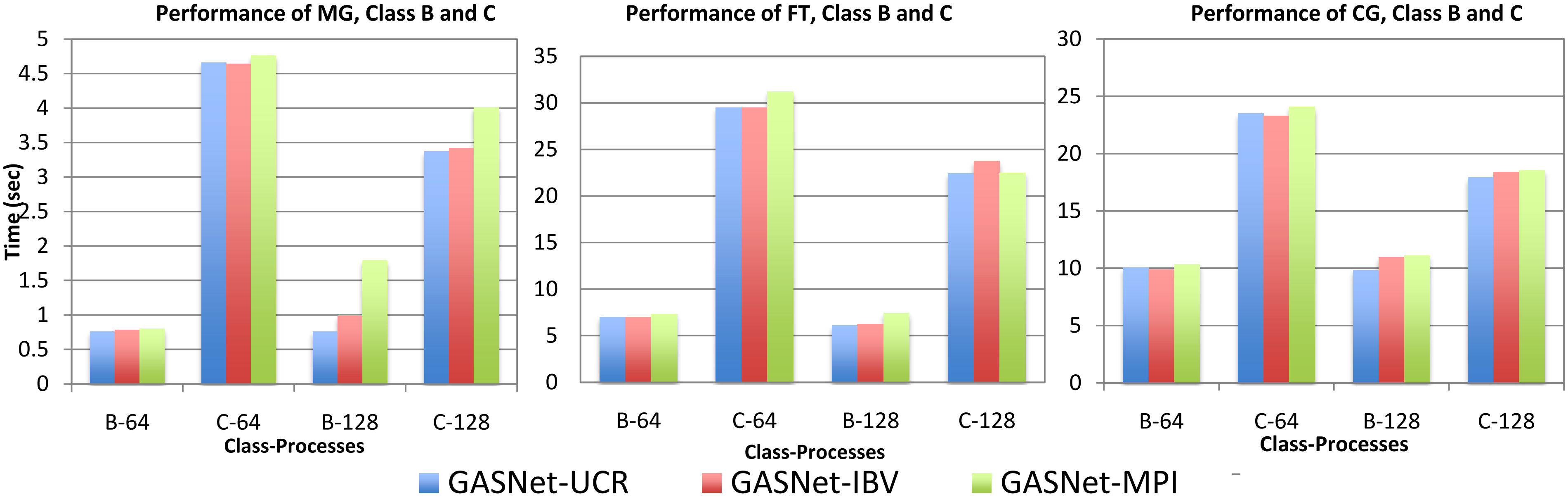
Multi-Endpoint: upc_memput





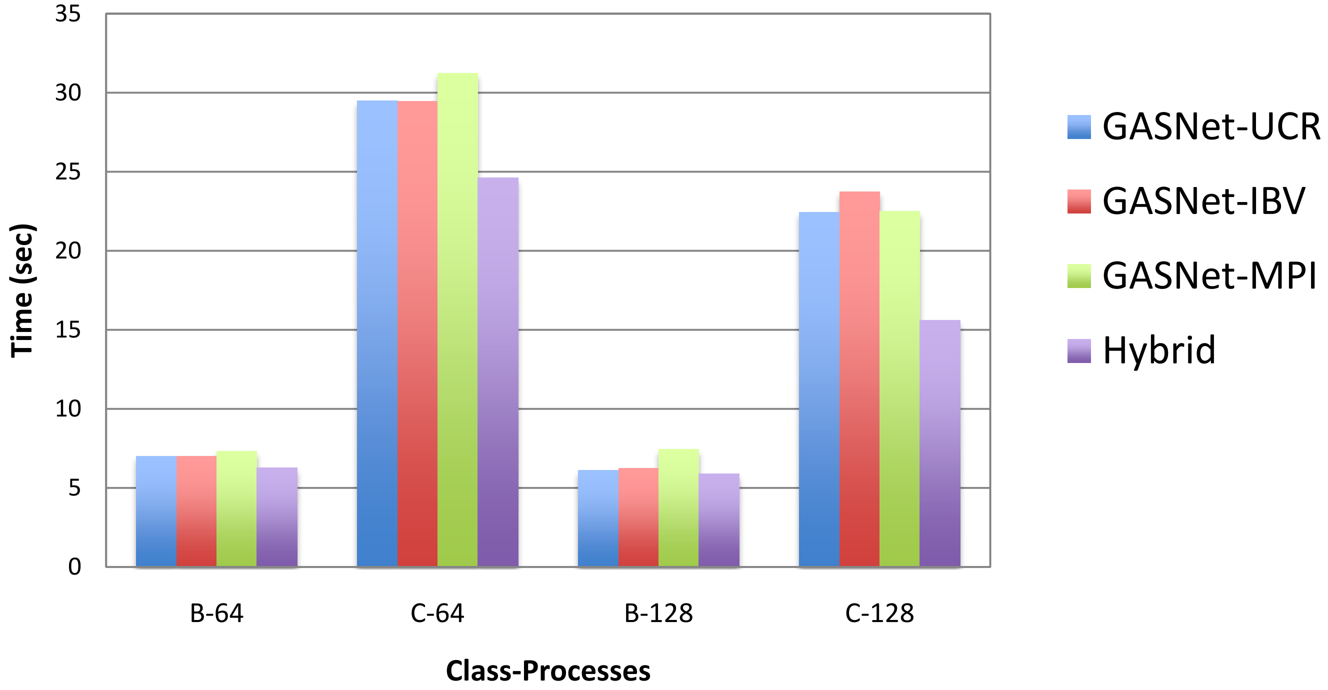
- Latency for multi-thread lowered by 80%
- For 1K message, bandwidth is increased from 72MB/s to 202MB/s

Evaluation using UPC NAS Benchmarks



- GASNet-UCR performs equal or better than GASNet-IBV
- 10% improvement for CG (B, 128)
- 23% improvement for MG (B, 128)

Evaluation of Hybrid MPI+UPC NAS-FT



- Modified NAS FT UPC all-to-all pattern using MPI_Alltoall
- Truly hybrid program
- 34% improvement for FT (C, 128)

Summary

- Unified Communication Runtime (UCR): supports MPI and UPC simultaneously on OFED
- Promising: MPI communication not harmed and UPC communication not penalized
- Pure UPC NAS: 10% improvement CG (B, 128), 23% improvement MG (B, 128)
- MPI+UPC FT: 34% improvement for FT (C, 128)
- Multi-endpoint version improves multi-threaded latency by 80%
- Allows to solve problems using multiple programming modes
  - MPI only, PGAS (UPC) only and hybrid (MPI and UPC)
- Suitable candidate for Exascale Computing

Thank You!

{panda, surs}@cse.ohio-state.edu

Network-Based Computing Laboratory

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

MVAPICH Web Page

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