Unified MPI and PGAS (UPC, OpenShmem, etc.) Design with RDMA to Support Hybrid Programming Environment for Exascale Systems

A Presentation at OFA Conference, Monterey 2012

by

Dhabaleswar K. (DK) Panda

The Ohio State University

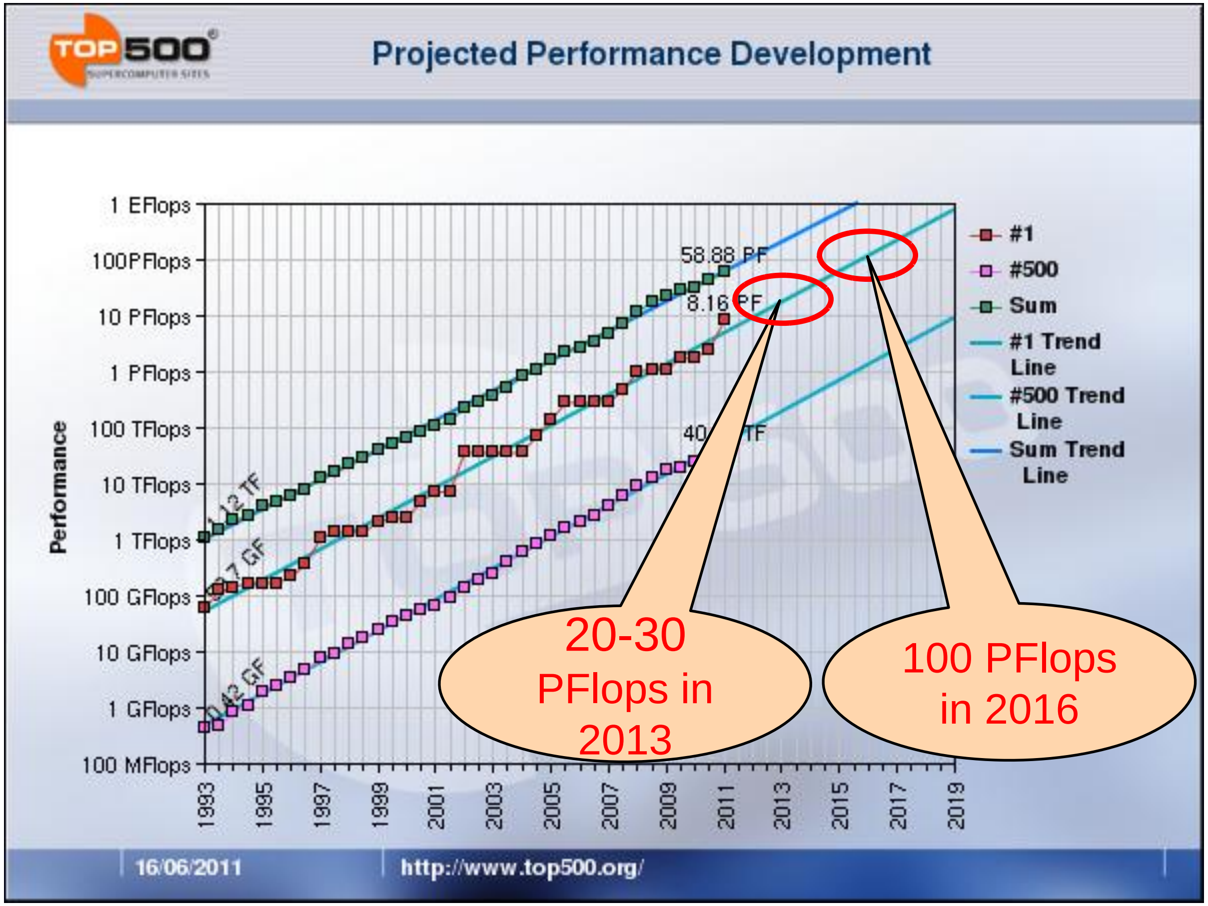
E-mail: panda@cse.ohio-state.edu

http://www.cse.ohio-state.edu/~panda

Outline

- Exascale Computing and Hybrid Programming Model
- Challenges in unifying UPC and MPI
- Solutions and Experimental Results
- Challenges in unifying MPI and OpenSHMEM
- Solutions and Experimental Results
- Conclusions

High-End Computing (HEC): PetaFlop to ExaFlop



Expected to have an ExaFlop system in 2019 -2020!

Exaflop Computing

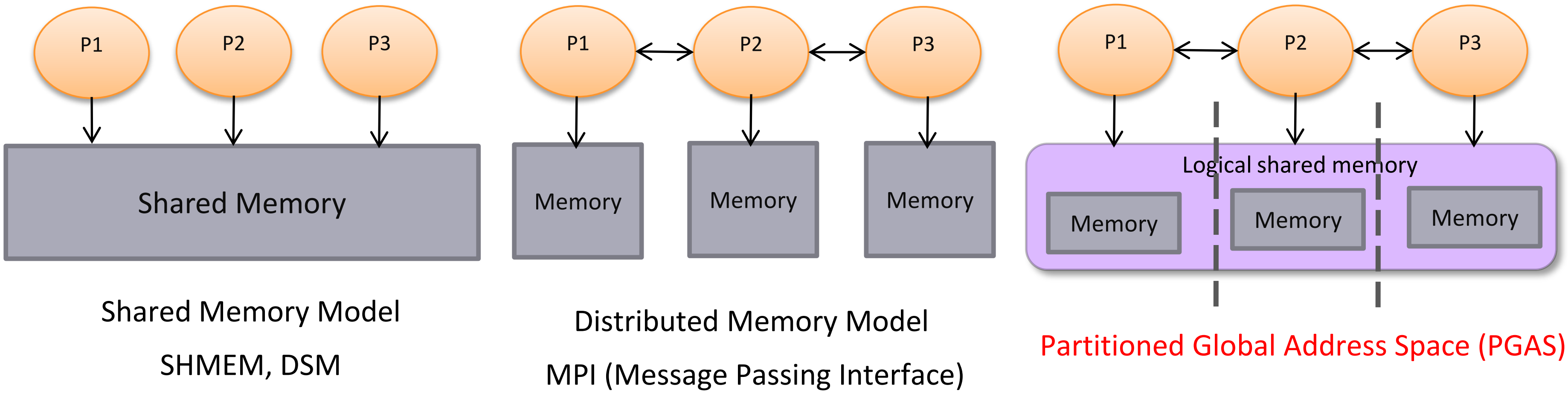
- Exaflop = 10^{18} floating point operations per second
- Represents a factor of 100-1000x from current state of the art
- Goal – Reach Exaflop levels by 2019-2020
- Exaflop computing is expected to spur research into high performance technologies
- Discover new technologies to enable next generation of science

Exascale System Targets

Systems	2010	2018	Difference Today & 2018
System peak	2 PFlop/s	1 EFlop/s	O(1,000)
Power	6 MW	~20 MW (goal)	
System memory	0.3 PB	32 – 64 PB	O(100)
Node performance	125 GF	1.2 or 15 TF	O(10) – O(100)
Node memory BW	25 GB/s	2 – 4 TB/s	O(100)
Node concurrency	12	O(1k) or O(10k)	O(100) – O(1,000)
Total node interconnect BW	3.5 GB/s	200 – 400 GB/s (1:4 or 1:8 from memory BW)	O(100)
System size (nodes)	18,700	O(100,000) or O(1M)	O(10) – O(100)
Total concurrency	225,000	O(billion) + [O(10) to O(100) for latency hiding]	O(10,000)
Storage capacity	15 PB	500 – 1000 PB (>10x system memory is min)	O(10) – O(100)
IO Rates	0.2 TB	60 TB/s	O(100)
MTTI	Days	O(1 day)	-O(10)

Courtesy: DOE Exascale Study and Prof. Jack Dongarra

Partitioned Global Address Space (PGAS) Models



- Global view improves programmer productivity
- Idea is to decouple data movement with process synchronization
- Processes should have asynchronous access to globally distributed data
- Well suited for irregular applications and kernels that require dynamic access to different data

Different Approaches for Supporting PGAS Models

- Library-based
  - Global Arrays
  - OpenSHMEM
- Compiler-based
  - Unified Parallel C (UPC)
  - Co-Array Fortran (CAF)
- HPCS Language-based
  - X10
  - Chapel
  - Fortress

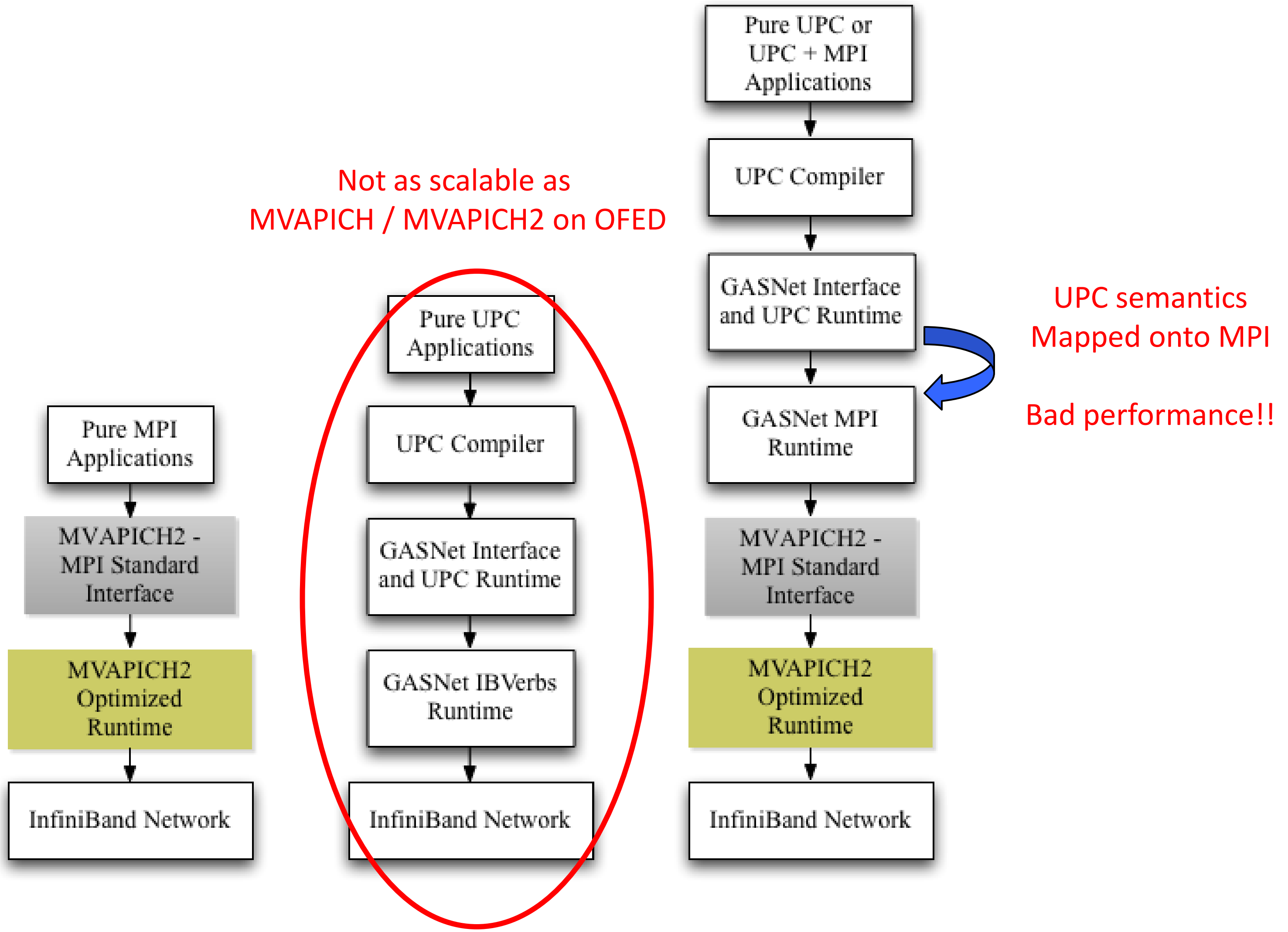
Outline

- Exascale Computing and Hybrid Programming Model
- Challenges in unifying UPC and MPI
- Solutions and Experimental Results
- Challenges in unifying MPI and OpenSHMEM
- Solutions and Experimental Results
- Conclusions

Issues and Problems

- Parts of big applications and third party libraries use MPI
- Parallel Math and Physics libraries have very high investment, ***cannot*** re-write them!
- Separate runtimes for MPI and UPC/OpenSHMEM ?
  - Requires more network resources
  - Must ensure progress of both MPI and UPC/OpenSHMEM runtimes
  - May even lead to deadlock!
  - Issues with performance and scalability
  - Don't interoperate very well
- No unified runtime to support both MPI and UPC/OpenSHMEM over OFED with best performance and scalability
  - Current performance comparison between MPI and UPC/OpenSHMEM is misleading
- No unified runtime to design hybrid programs (MPI+UPC or MPI+OpenSHMEM) on emerging multi-core environments

Various ways to use UPC and MPI and Limitations



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 with RDMA
  - 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

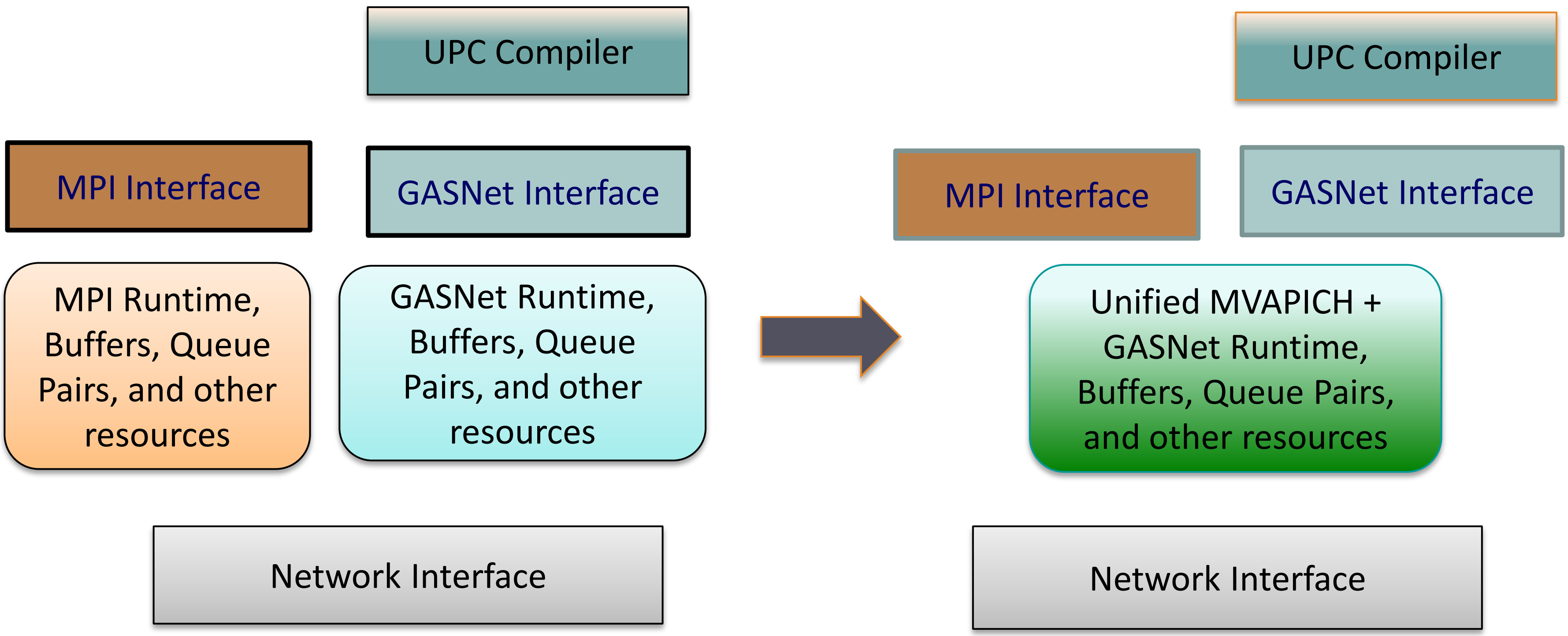
- Allow scientists to develop applications in the following modes
  - MPI only
  - PGAS (UPC) only
  - Hybrid (MPI and UPC)

- Allow scientists to evaluate the impact of programming models on applications on next generation systems in a fair manner

Outline

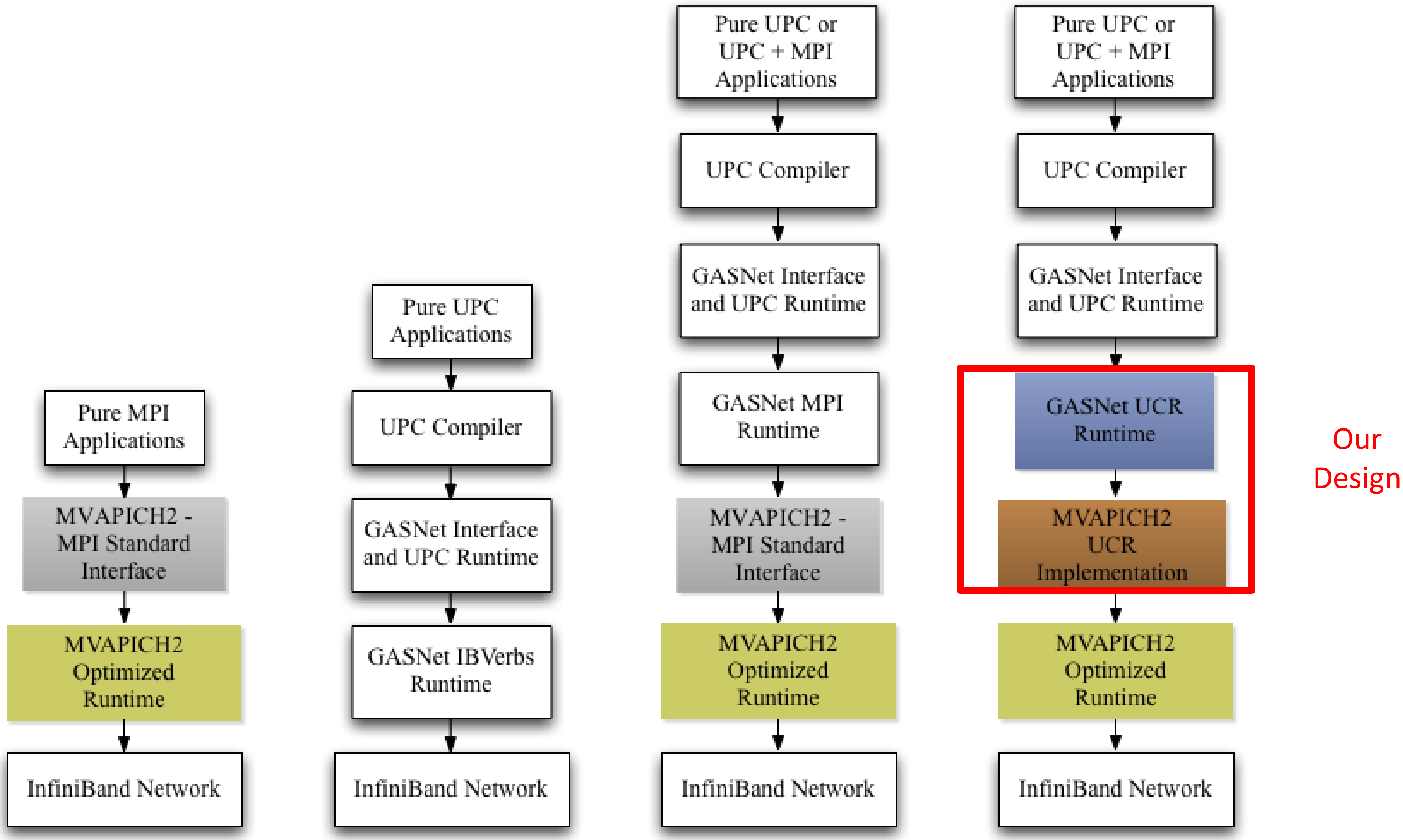
- Exascale Computing and Hybrid Programming Model
- Challenges in unifying UPC and MPI
- Solutions and Experimental Results
- Challenges in unifying MPI and OpenSHMEM
- Solutions and Experimental Results
- Conclusions

Unifying UPC and MPI Runtimes: Experience with MVAPICH2

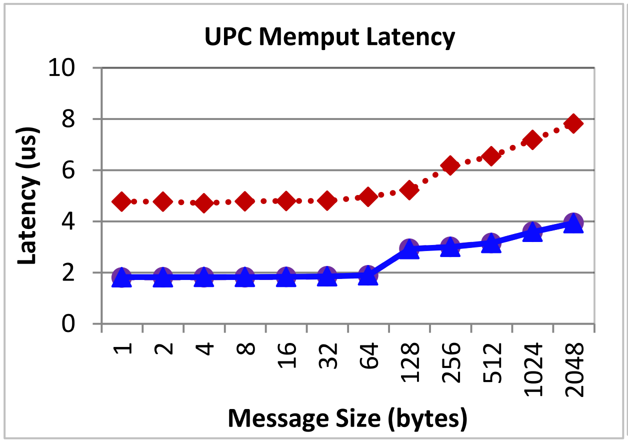


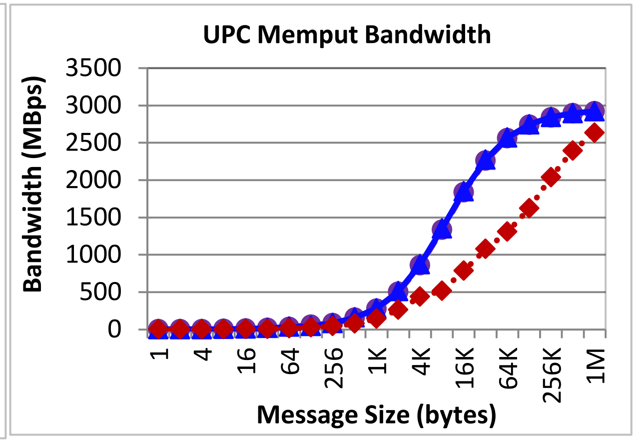
- Currently UPC and MPI do not share runtimes
  - Duplication of lower level communication mechanisms
  - GASNet unable to leverage advanced buffering mechanisms developed for MVAPICH2
- Our novel approach is to enable a truly unified communication library

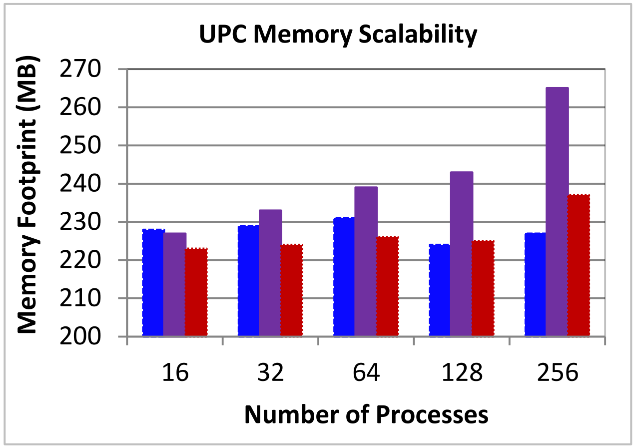
New Configuration for UPC and MPI



UPC Micro-benchmark Performance





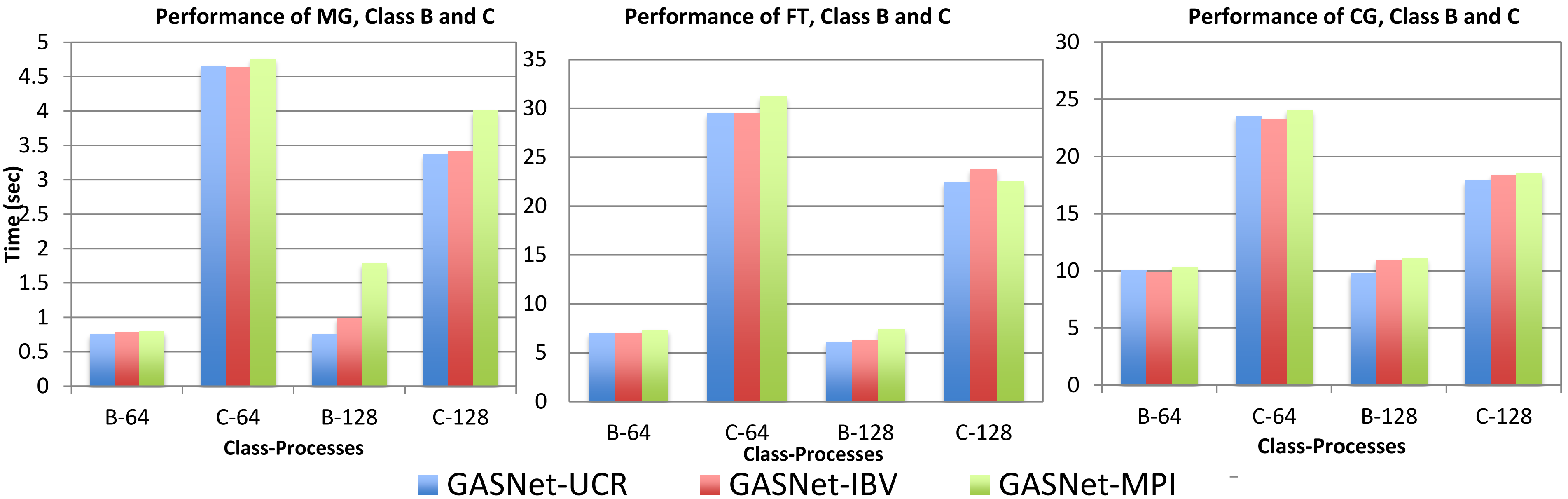


GASNet-UCR GASNet-IBV GASNet-MPI

- BUPC micro-benchmarks from latest release 2.10.2
- UPC performance is identical with both native IBV layer and new UCR layer
- Performance of GASNet-MPI conduit is not very good
  - Mismatch of MPI specification and Active messages
- GASNet-UCR is more scalable compared native IBV conduit

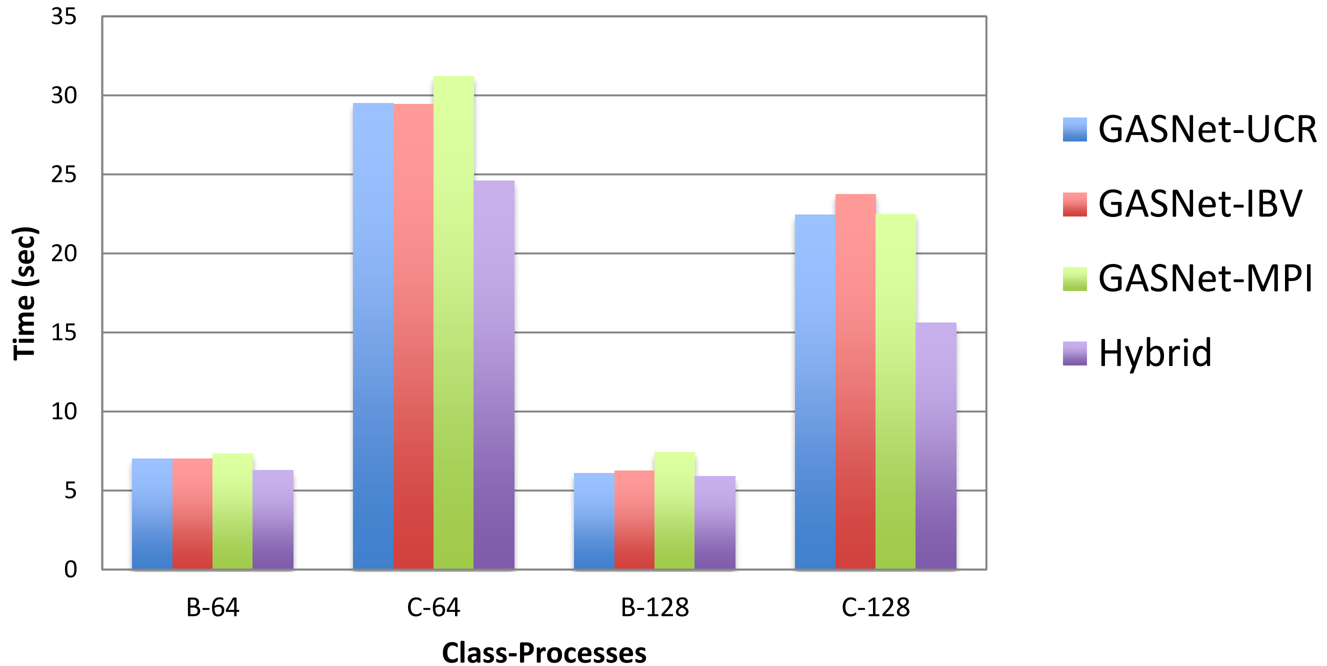
J. Jose, M. Luo, S. Sur and D. K. Panda, “Unifying UPC and MPI Runtimes: Experience with MVAPICH”, International Conference on Partitioned Global Address Space (PGAS), 2010

Evaluation using UPC NAS Benchmarks



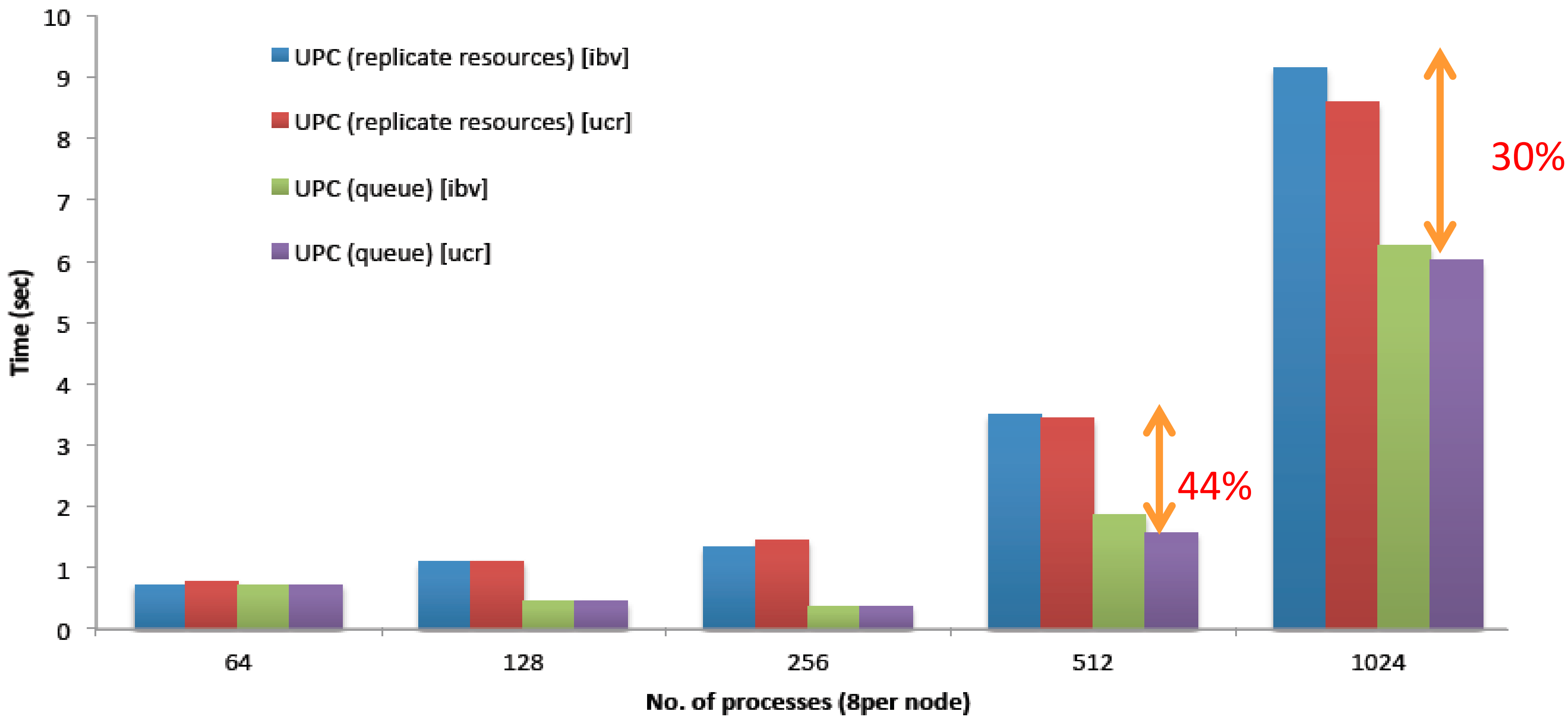
- GASNet-UCR performs equal or better than GASNet-IBV
- 10% improvement for CG (B, 128)
- 23% improvement for MG (B, 128)

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)

Graph500 Results with new UPC Queue Design



- Workload – Scale:24, Edge Factor:16 (16 million vertices, 256 million edges)
- 44% Improvement over base version for 512 UPC-Threads
- 30% Improvement over base version for 1024 UPC-Threads

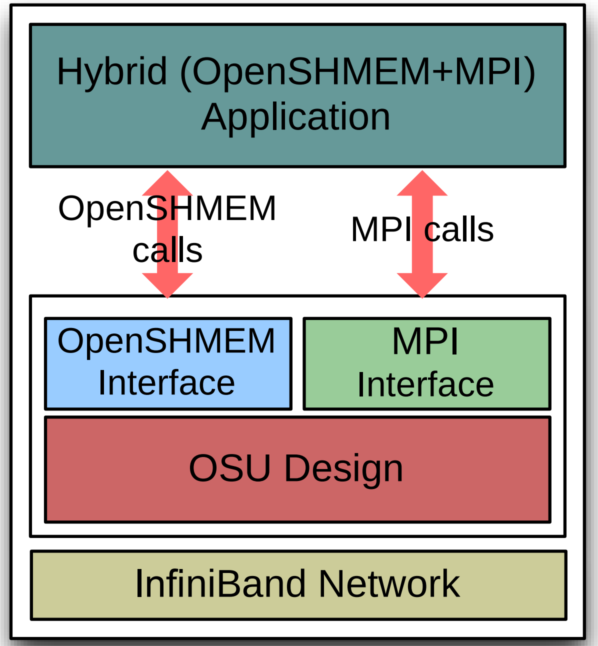
J. Jose, S. Potluri, M. Luo, S. Sur and D. K. Panda, UPC Queues for Scalable Graph Traversals: Design and Evaluation on InfiniBand Clusters, Fifth Conference on Partitioned Global Address Space Programming Model (PGAS '11), Oct. 2011.

Outline

- Exascale Computing and Hybrid Programming Model
- Challenges in unifying UPC and MPI
- Solutions and Experimental Results
- Challenges in unifying MPI and OpenSHMEM
- Solutions and Experimental Results
- Conclusions

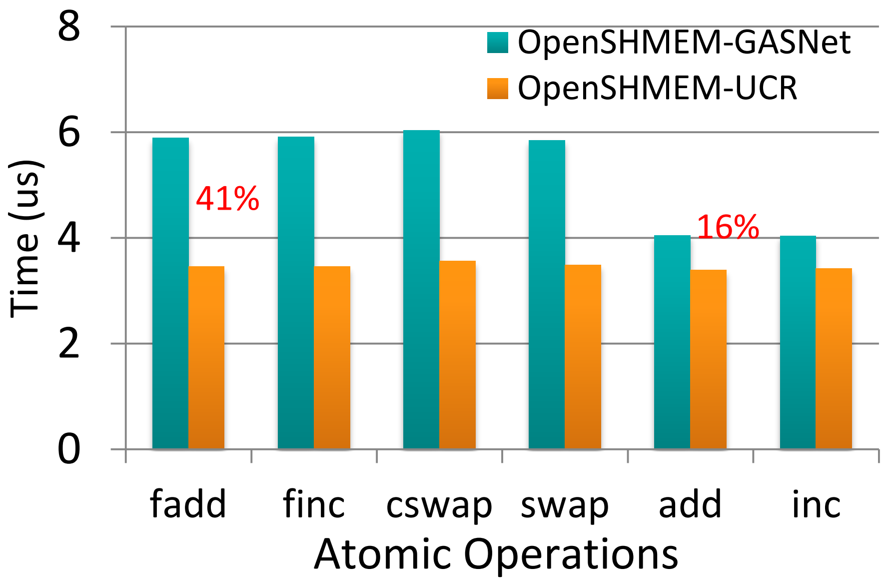
Scalable OpenSHMEM and Hybrid (MPI and OpenSHMEM) designs

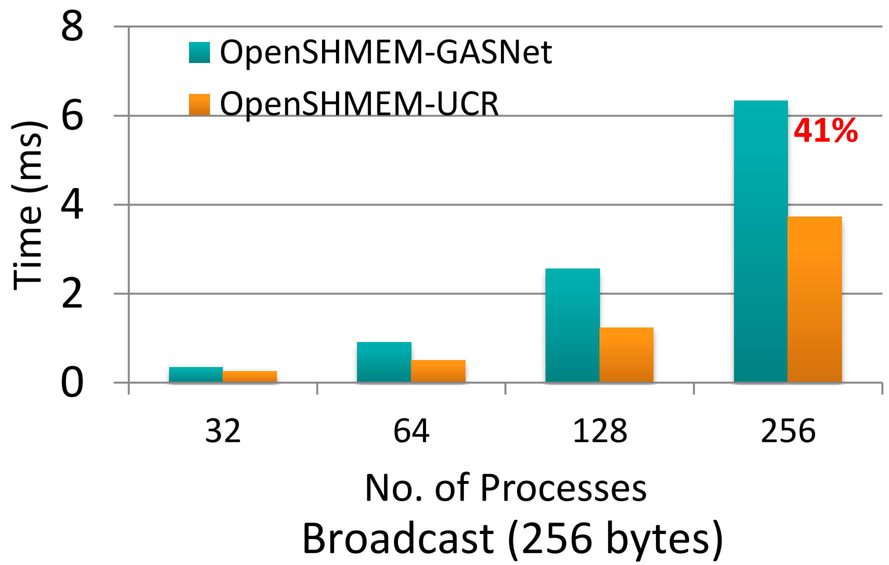
- Based on OpenSHMEM Reference Implementation http://openshmem.org/
  - Provides a design over GASNet
  - Does not take advantage of all OFED features
- Design scalable and High-Performance OpenSHMEM over OFED
- Designing a Hybrid MPI +OpenSHMEM Model
  - Current Model – Separate Runtimes for OpenSHMEM and MPI
    - Possible deadlock if both runtimes are not progressed
    - Consumes more network resource
  - Our Approach – Single Runtime for MPI and OpenSHMEM

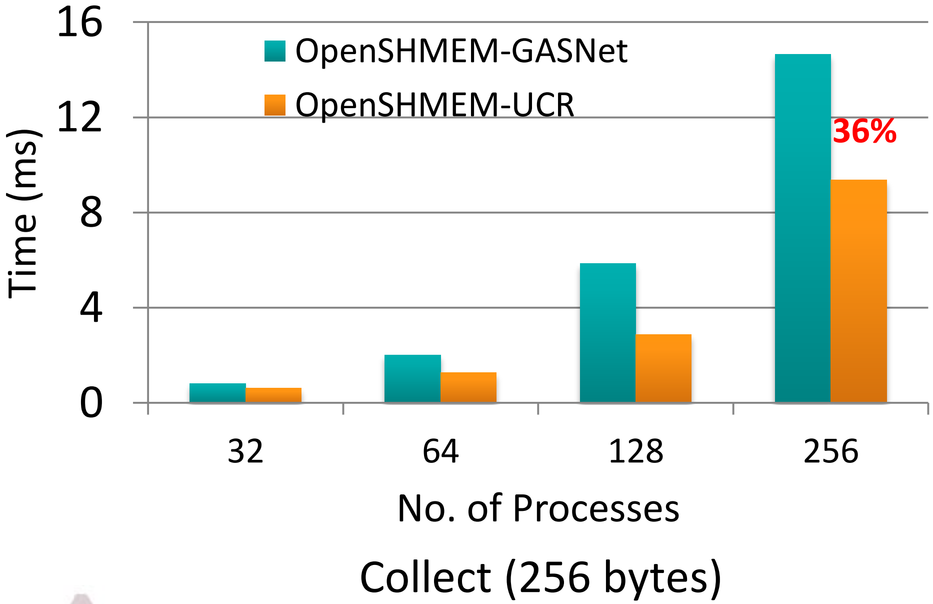


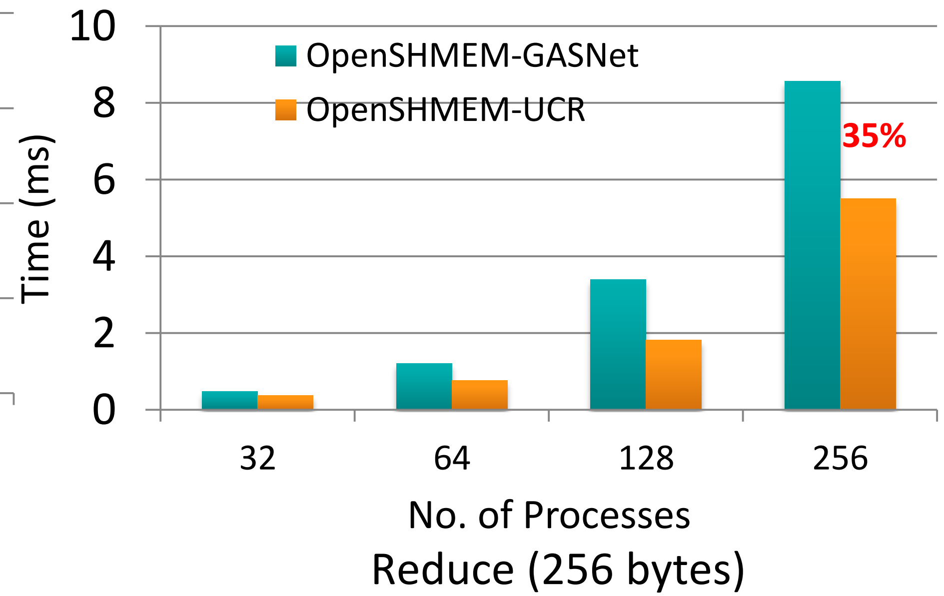
Hybrid MPI+OpenSHMEM

Micro-Benchmark Performance (OpenSHMEM)

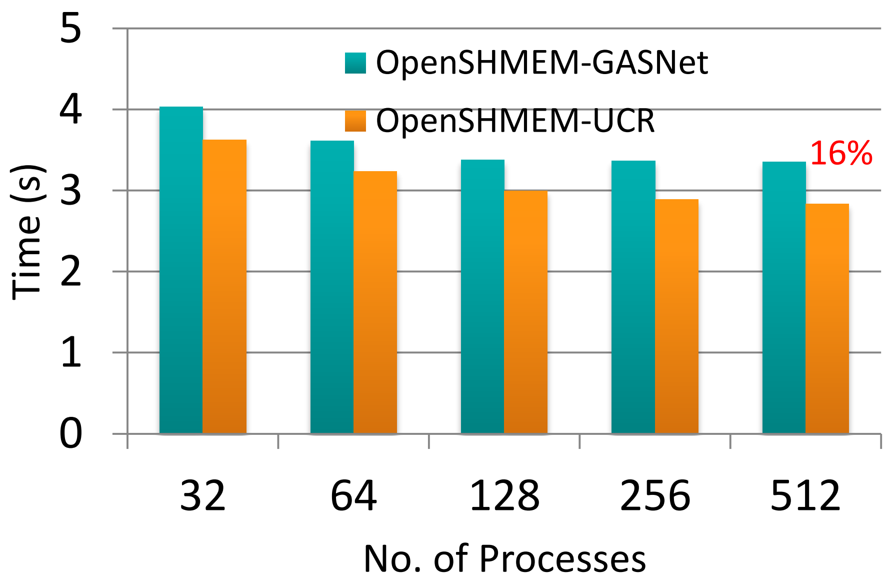




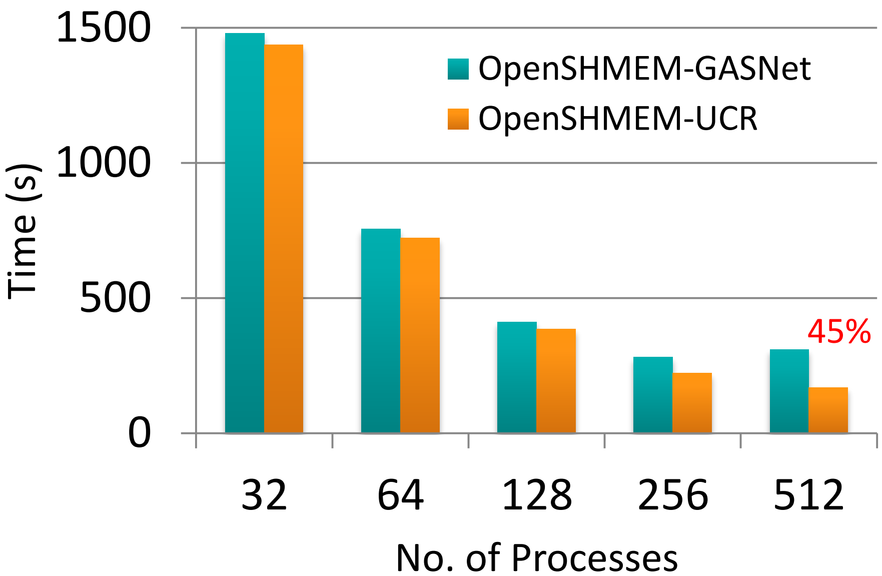




Performance of OpenSHMEM Applications



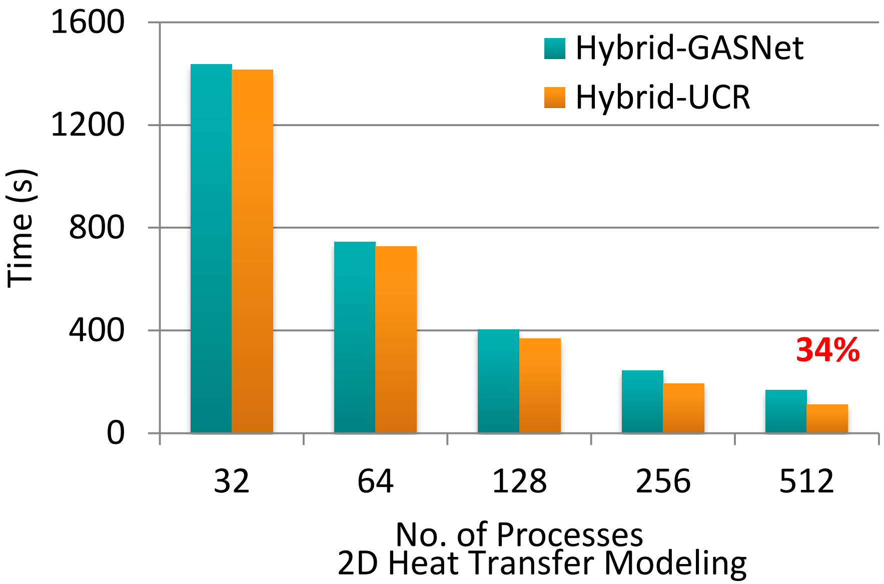
2D FFT

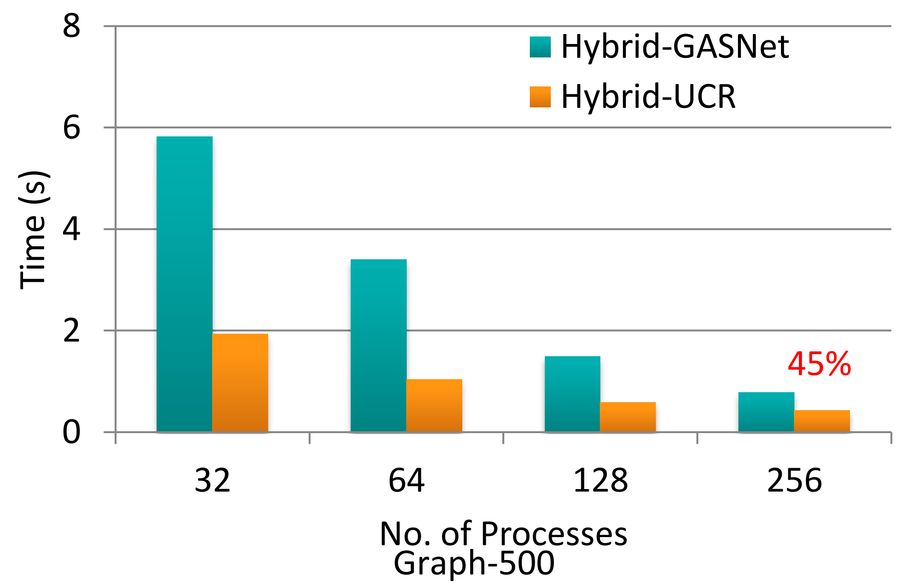


2D Heat Transfer Modeling

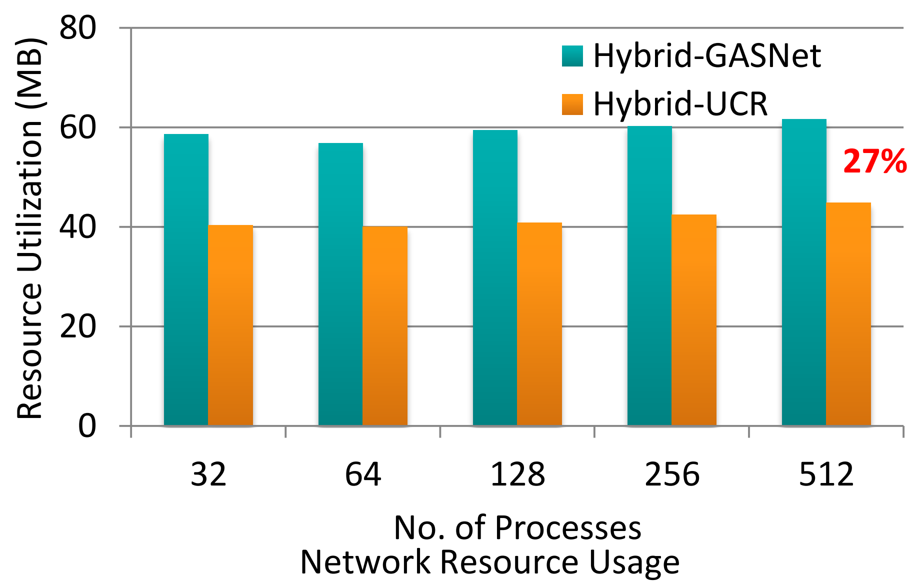
- 2D FFT with 8K input matrix
    - 16% improved performance for 512 processes
- 2D Heat Transfer Modeling
    - 45% improved performance for 512 processes
- Performance Improvement because of high performance runtime

Performance of Hybrid (OpenSHMEM+MPI) Applications





- Improved Performance for Hybrid Applications
  - 34% improvement for 2DHeat Transfer Modeling with 512 processes
  - 45% improvement for Graph500 with 256 processes
- Our approach with single Runtime consumes 27% lesser network resources



Conclusions

- Hybrid programming models are critical for Exascale systems
- Unified Communication Runtime (UCR)
  - Supports MPI+UPC and MPI+OpenSHMEM simultaneously on OFED using RDMA features
- Promising:
  - MPI communication not harmed
  - {UPC, OpenSHMEM} communication performance and scalability are improved
- Allows to solve problems using multiple programming modes
  - MPI only
  - PGAS (UPC) only
  - PGAS (OpenSHMEM)
  - Hybrid (MPI and UPC)
  - Hybrid (MPI and OpenSHMEM)
- Suitable candidate for Exascale Computing

Web Pointers

http://www.cse.ohio-state.edu/~panda

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

MVAPICH Web Page

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

MVAPICH

panda@cse.ohio-state.edu