Scaling with PGAS Languages

Panel Presentation at OFA Developers Workshop (2013)

by

Dhabaleswar K. (DK) Panda

The Ohio State University

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

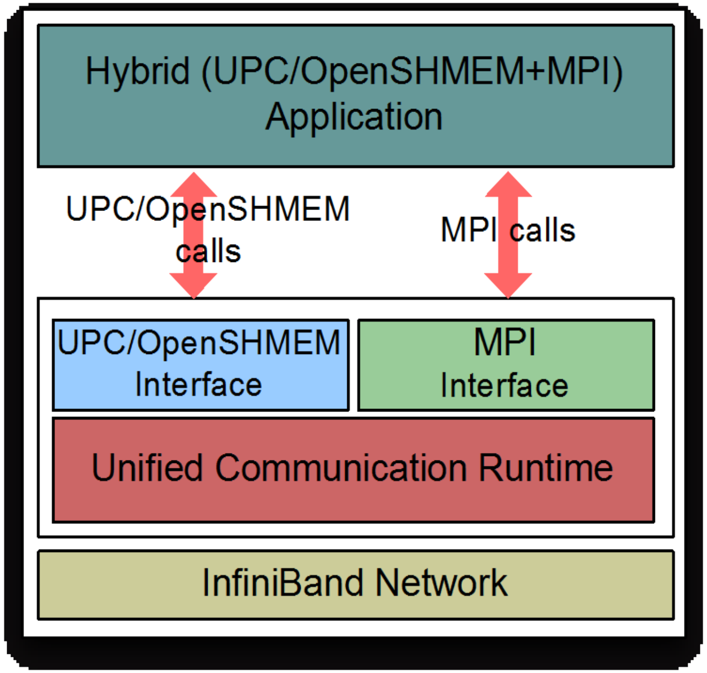
http://www.cse.ohio-state.edu/~panda

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
    - 17th ranked 73,278-core cluster (Tsubame 2.0) at Tokyo Institute of Technology
    - 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

Overview of MVAPICH2-X

- Can support the following programming models over OFA verbs
  - PGAS
    - UPC
    - OpenSHMEM
  - MPI (with OpenMP)
  - Hybrid (MPI and PGAS)
    - MPI (w/ OpenMP) + UPC
    - MPI (w/ OpenMP) + OpenSHMEM

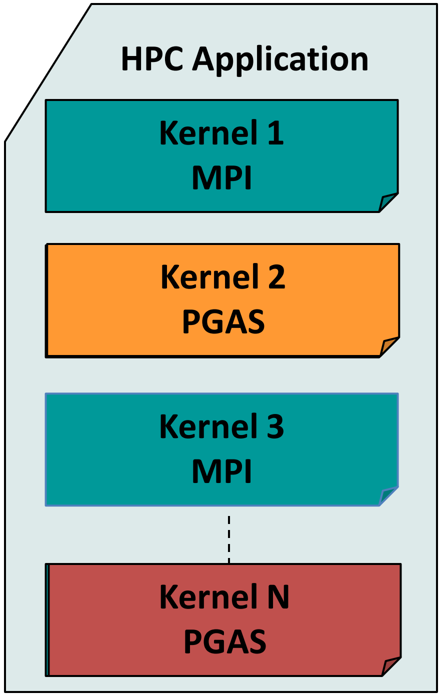


- Unified communication runtime allows flexible support for all these programming models
- Can be downloaded from

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

Support for Flexible Hybrid (MPI+PGAS) Programming

- Application sub-kernels can be re-written in MPI/PGAS based on communication characteristics
- Benefits:
  - Best of Distributed Computing Model
  - Best of Shared Memory Computing Model
- Exascale Roadmap*:
  - "Hybrid Programming is a practical way to program exascale systems"



* The International Exascale Software Roadmap, Dongarra, J., Beckman, P. et al., Volume 25, Number 1, 2011, International Journal of High Performance Computer Applications, ISSN 1094-3420

PGAS Models

Q: Shared Memory Models: **“Of the models for distributed computing, what in your view is the significance of the recent emergence of PGAS languages?”**

- PGAS models improve programmability
- Can improve performance of irregular applications
- Hybrid Programming models allow incremental application development using MPI+PGAS models

PGAS Runtime Implementation

Q: Implementing PGAS: **"Each of you has looked at various implementations of interfaces for PGAS languages. How have you implemented the interface, and what has your experience been with it to date?"**

- Runtimes should provide flexibility to choose between PGAS and Message Passing semantics
- Runtimes for PGAS or Message Passing models have to address a core set of issues
- Critical to efficiently use network and memory resources
- MVAPICH2-X provides a unified runtime for hybrid MPI+PGAS models, offers deadlock-free communication progress across models, better performance and optimal network resource usage
- MVAPICH2-X UPC/OpenSHMEM bindings are implemented over active messages, one-sided operations, and atomic/synchronizations operations

Memory Consistency and Protection

Q: memory consistency: **“UPC has a well defined memory consistency model governing the reading and writing characteristics of shared memory. What aspects of RDMA-capable networks have made conformity to this memory consistency model particularly challenging for UPC compilers?”**

- UPC offers `strict’ and `relaxed’ modes
- Runtime can use RDMA completion events for implementing consistency modes

Q: Memory Protection: **“Current IB architecture defines a system of memory keys which are exchanged between communicating partners. Is this an appropriate model to be used in PGAS implementations?”**

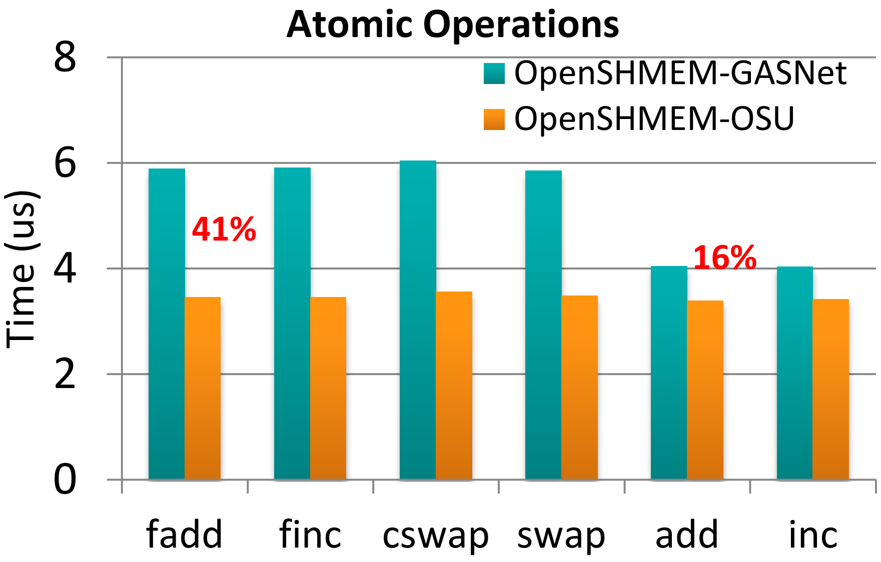
- Registration cache in MVAPICH2-X alleviates registration costs
- Can register symmetric memory regions at initialization

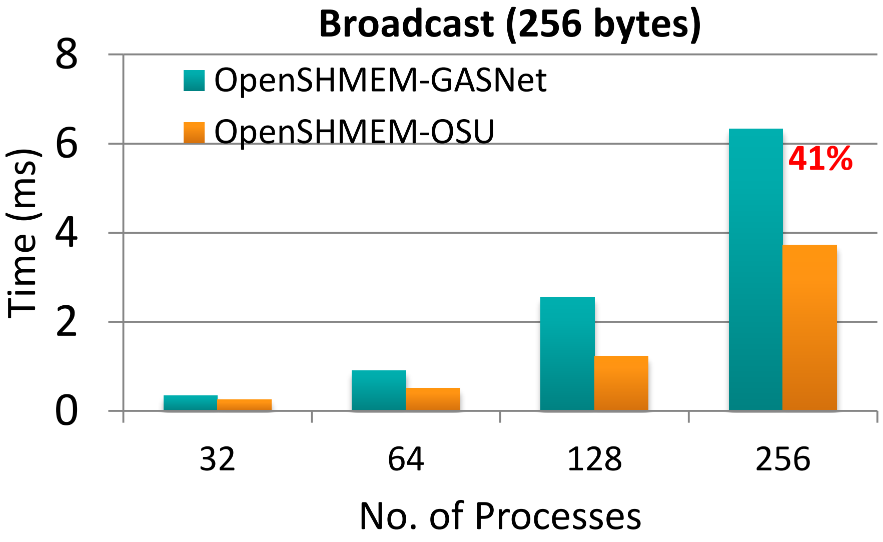
Thread Safety in PGAS Runtime

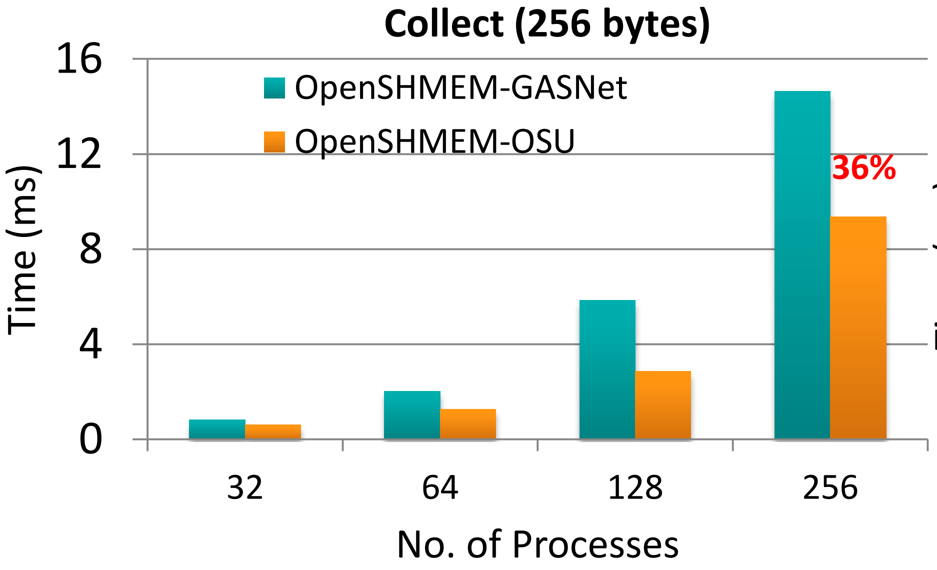
Q: thread safety: **"How important is it for a PGAS compiler that the API it uses for accessing the RDMA-capable network be thread safe?"**

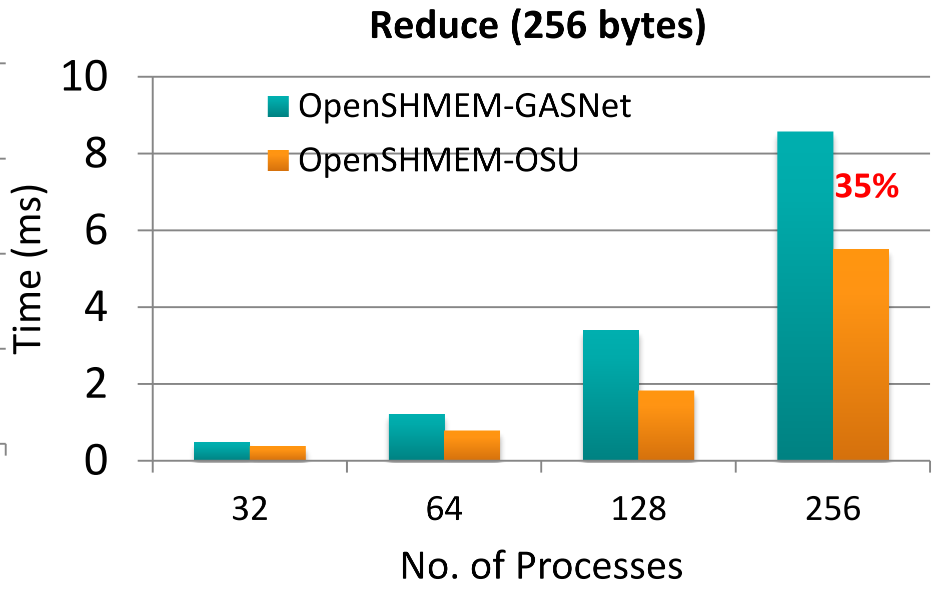
- Multi-end point design can enable thread-safety
- The multi-endpoint design offers more freedom to compiler
- Performance benefits with Multi-threaded Multi-Network Endpoint Runtime for UPC
  - M. Luo, J. Jose, S. Sur, and D. K. Panda, Multithreaded UPC Runtime with Network Endpoints: Design Alternatives and Evaluation on Multi-core Architectures, High Performance Computing (HiPC'11), December 2011

Micro-Benchmark Performance (OpenSHMEM)

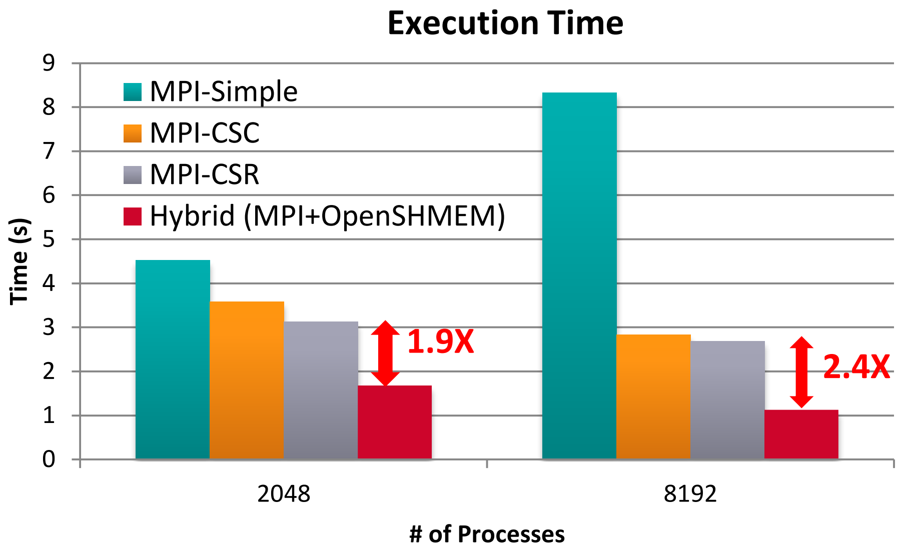


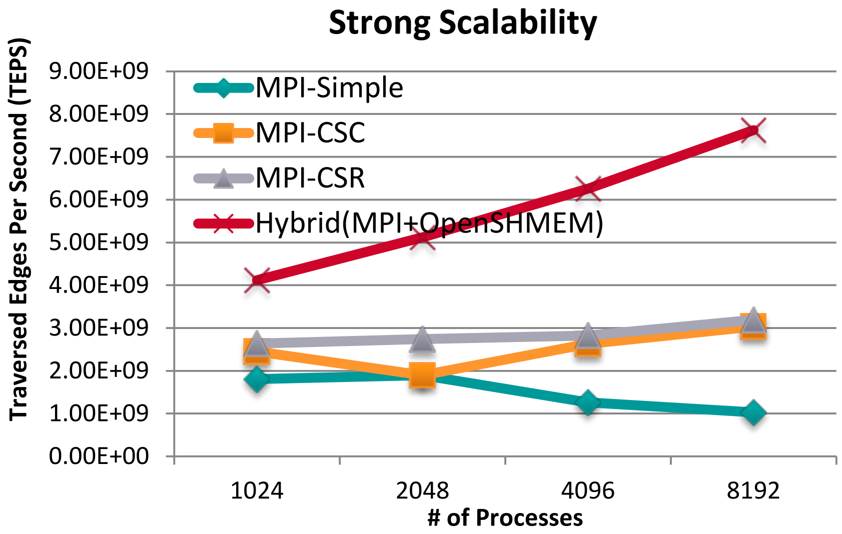




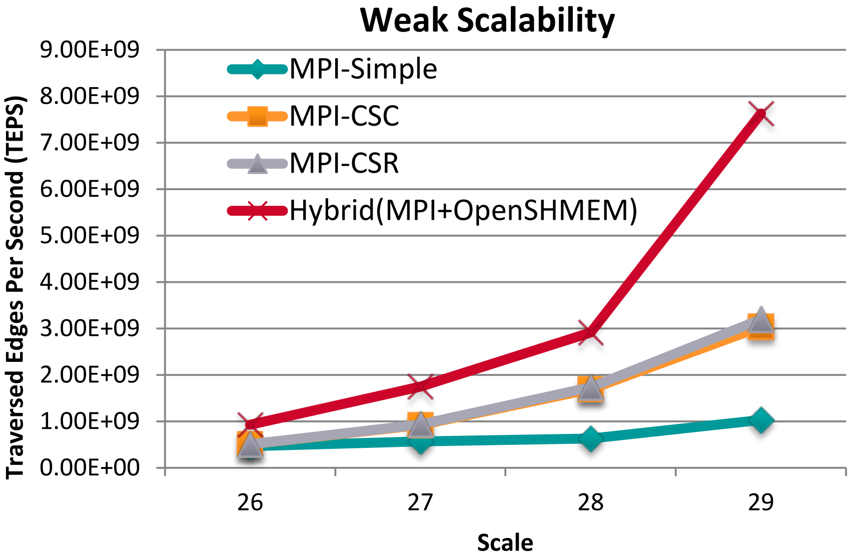


Hybrid MPI+OpenSHMEM Graph500 Design





- Performance of Hybrid (MPI+OpenSHMEM) Graph500 Design
  - 2,048 processes
    - **1.9X** improvement over MPI-CSR (best performing MPI version)
    - **2.7X** improvement over MPI-Simple (same communication characteristics)
  - 8,192 processes
    - **2.4X** improvement over MPI-CSR
    - **7.6 X** improvement over MPI-Simple



J. Jose, S. Potluri, K. Tomko and D. K. Panda, Designing Scalable Graph500 Benchmark with Hybrid MPI+OpenSHMEM Programming Models, International Supercomputing Conference (ISC'13), June 2013