Scalable MiniMD Design with Hybrid MPI and OpenSHMEM

Mingzhe Li[1], Jian Lin[1], Xiaoyi Lu[1], Khaled Hamidouche[1], Karen Tomko[2], Dhabaleswar K. (DK) Panda[1]

[1]Network-Based Computing Laboratory
Department of Computer Science and Engineering
The Ohio State University, USA

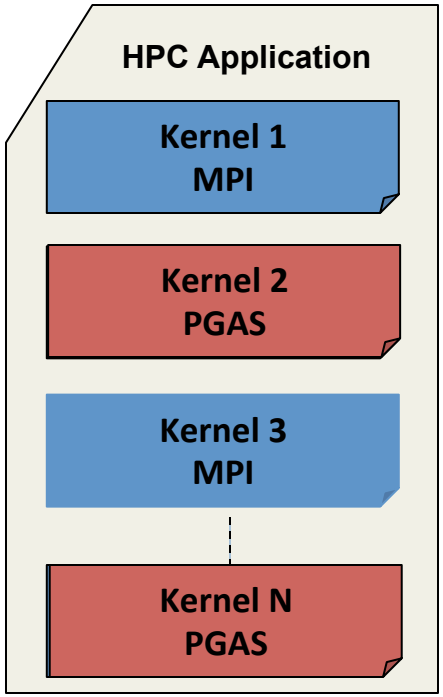
[2]Ohio Supercomputer Center, USA

Outline

- Introduction & Problem Statement
- Designing a Hybrid MPI and OpenSHMEM MiniMD
- Performance Evaluation
- Conclusion & Future Work

Hybrid (MPI+PGAS) Programming for Exascale Systems

- 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

Overview of MiniMD

- MiniMD is a Molecular Dynamics (MD) mini-application in the Mantevo project at Sandia National Laboratories
- It has a stencil communication pattern which employs point-to-point message passing with irregular data
- Primary work loop inside MiniMD
  - Migrate the atoms to different ranks in every 20th iteration
  - Exchange position information of atoms in boundary regions
  - Compute forces based on local atoms and those in boundary region from neighboring ranks
  - Exchange force information of atoms in boundary regions
  - Update velocities and positions of local atoms

Problem Statement

- *Can Hybrid MPI and OpenSHMEM be used to design high-performance and scalable MiniMD?*

- *What are the performance benefits that can be achieved by using Hybrid design on modern HPC clusters?*

Outline

- Introduction & Problem Statement
- Designing a Hybrid MPI and OpenSHMEM MiniMD
- Performance Evaluation
- Conclusion & Future Work

Design Challenges

- Synchronization between origin and target processes
  - How does the origin process know where to write data
  - How does the target process know when data has arrived
- Receive Buffer
  - Size of receive buffer
  - Memory scalability
- MVAPICH2/MVAPICH2-X Software
  - High Performance open-source MPI library for InfiniBand, 10Gig/iWARP, and RDMA over Converged Enhanced Ethernet (RoCE)
  - Used by more than 2,200 organizations in 73 countries
  - http://mvapich.cse.ohio-state.edu

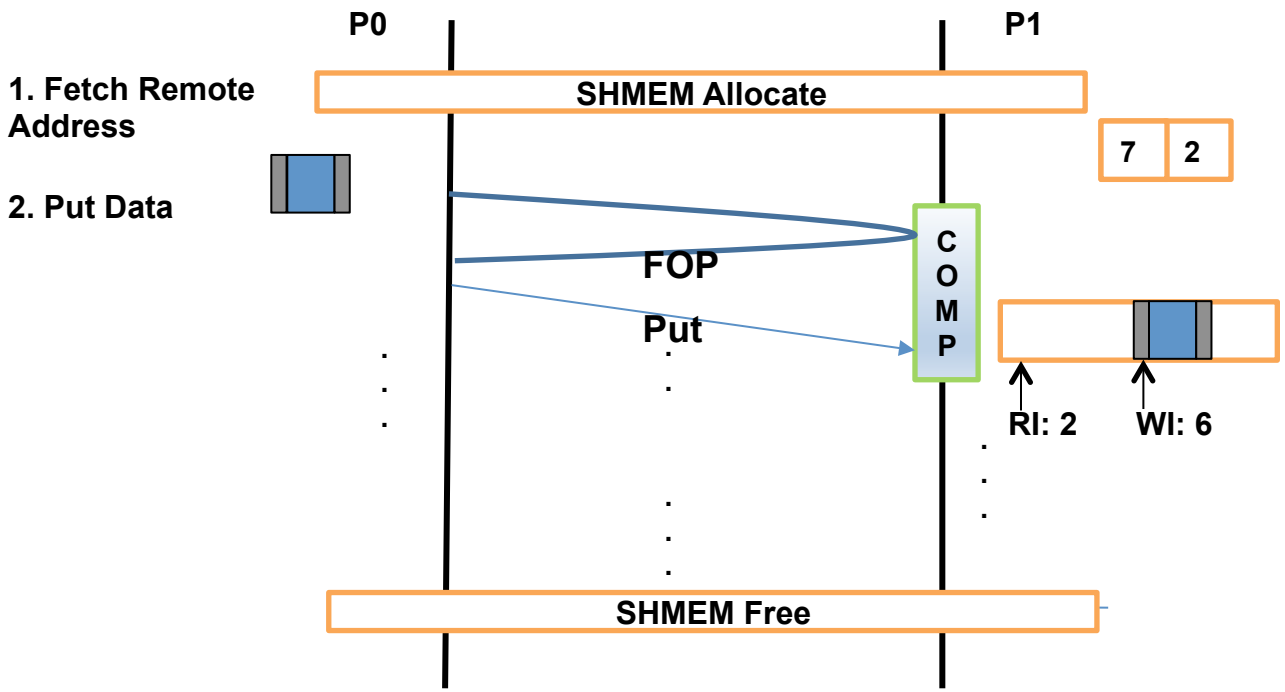
Design Alternatives: Hybrid-Barrier

- Modify existing MPI based version with OpenSHMEM routines
- Receive Buffer
  - Same receive buffer as that used for MPI version
  - Allocate receive buffer in OpenSHMEM heap
- Communication
  - Replace MPI Isend/Irecv with one-sided Put communication
  - Direct one-sided communication without interrupting target process
- Synchronization
  - OpenSHMEM barrier before and after Put

Design Alternatives: Hybrid-Advanced

- Better buffer management than Hybrid-Barrier design and remove barrier
- Receive Buffer
  - Receive buffer is in OpenSHMEM heap
  - Cyclic buffer management with write and read index
- Communication
  - Replace MPI Isend/Irecv with one-sided Put communication
- Synchronization
  - Use Atomic operation to get target address from target process
  - Target process polls on local receive buffer to ensure arrival of completion data

Overview of Hybrid-Advanced



- Each process allocates receive buffer
- Use Atomic operation to get target address
- Use Put to write data into receiver buffer

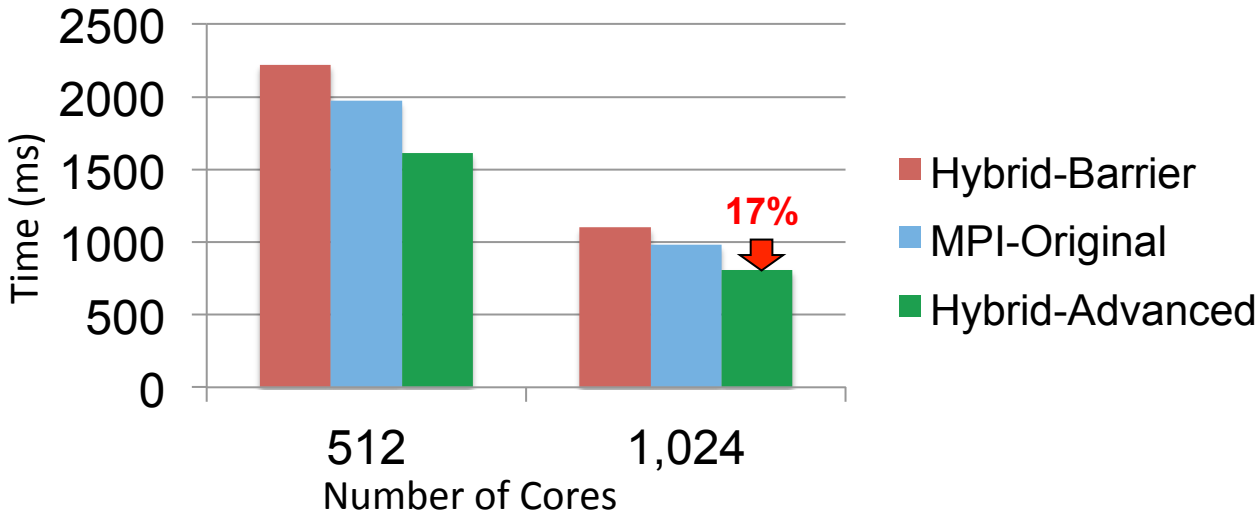
Outline

- Introduction & Problem Statement
- Designing a Hybrid MPI and OpenSHMEM MiniMD
- Performance Evaluation
- Conclusion & Future Work

Experiment Setup

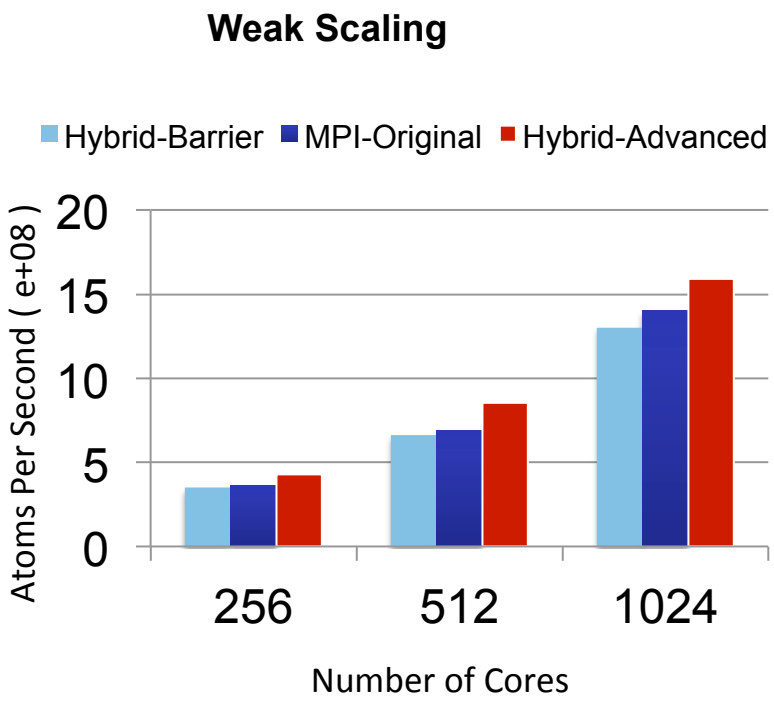
- TACC Stampede
  - Intel Sandybridge series of processors using Xeon dual 8 core sockets (2.70GHz) with 32GB RAM
  - Each node is equipped with FDR ConnectX HCAs (54 Gbps data rate) with PCI-Ex Gen3 interfaces
- OSU RI
  - Xeon dual 8 core sockets (2.67GHz) with 12GB RAM
  - Mellanox QDR ConnectX HCAs (32 Gbps data rate) with PCI-Ex Gen2 interfaces
- Software stack
  - MVAPICH2-X 2.0, MiniMD 1.0

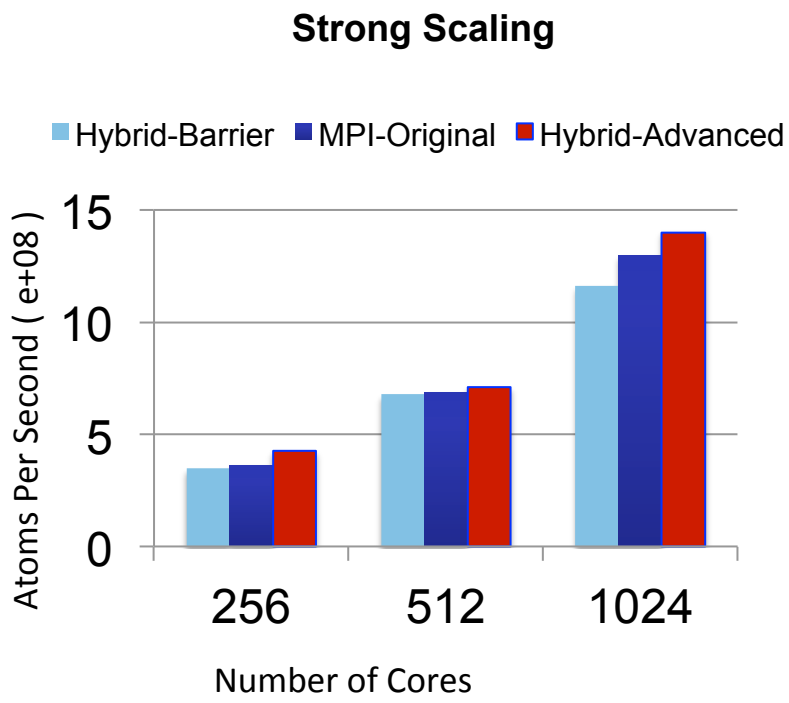
Total Execution Time



- On OSU RI
- Proposed Hybrid-Advanced design performs better than existing MPI implementation
- At 1,024 cores:
  - Hybrid–Advanced design improves by 17% over MPI based version

Scalability Results





- On TACC Stampede
- Input: 128 * 128 * 128 (Strong Scaling), 8192 grid cells per core (Weak Scaling)
- Hybrid-Advanced design shows good strong and weak scaling
- Hybrid-Advanced design scales better than existing implementations

Conclusion & Future Work

- Proposed a scalable MiniMD design with MPI and OpenSHMEM
- At 1K cores, proposed design achieves 17% improvement over MPI based implementation
- Hybrid-Advanced design shows good scalability characteristic
- Plan to re-design other applications with MPI and OpenSHMEM

Thank You!

{limin, linjia, luxi, hamidouc, panda}
@cse.ohio-state.edu

{Ktomko}@osc.edu

MVAPICH Web Page
http://mvapich.cse.ohio-state.edu/

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