MPI Advance – Optimizations and Extensions to MPI

Amanda Bienz, Derek Schafer, Patrick Bridges, University of New Mexico

Anthony Skjellum, Riley Shipley, Tennessee Tech

Purushotham V. Bangalore, University of Alabama

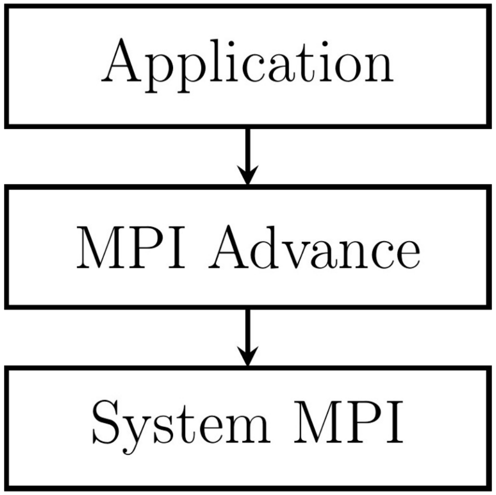
Center for Understandable, Performant Exascale Communication Systems

CUP-ECS Center Overview

- **Mission**: Provide optimized, performance-transparent communication systems for NNSA Exascale applications
- **Goal**: Research, demonstrate, and deploy better communication abstractions that make NNSA mission applications faster, more predictable, and easier to write
- **Center Leadership**
  - Patrick Bridges, University of New Mexico
  - Puri Bangalore, University of Alabama
  - Tony Skjellum, Tennessee Tech

Motivation

- A collection of lightweight libraries that sit on top of MPI
- Five main motivations:
  - Early access to new MPI functionality before MPI implementations
  - Conceptual prototyping + sharing of pre-standard features
  - Run new on older MPIs (for some cases)
  - Optimizations of existing APIs
  - Generate/support new and improved community best practices
- Middle ground enables optimization opportunities from each end
  - Applications adapt code to new APIs
  - Implementations get implementation knowledge
- Repository and framework to drive/motivate/validate
  - Next-generation of MPI standards (motivate standardization)
  - Bridge between parallel programming environments where MPI Forum may not ever choose to standardize (MPI+X and beyond)

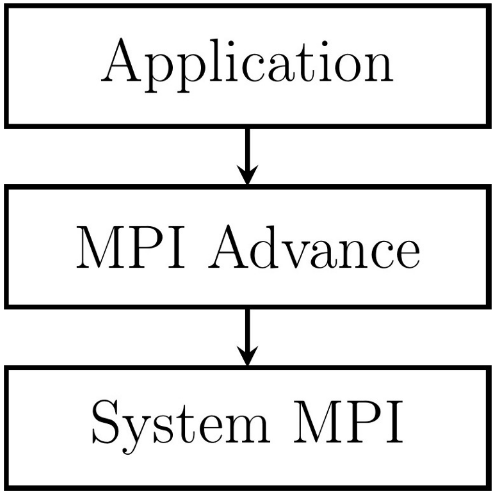

Application

MPI Advance

System MPI

Current Feature Set

- Partitioned Point-to-Point Operations (MPIPCL)
  - Supports MPI-4.0 for MPI implementations that don't yet have this functionality
  - Enables experimentation with different strategies not considered in a given MPI's partitioned communication implementation
- Locality Aware Collectives
  - New and better algorithms for topology-aware collective operations
  - Considers both CPU and GPU deployments

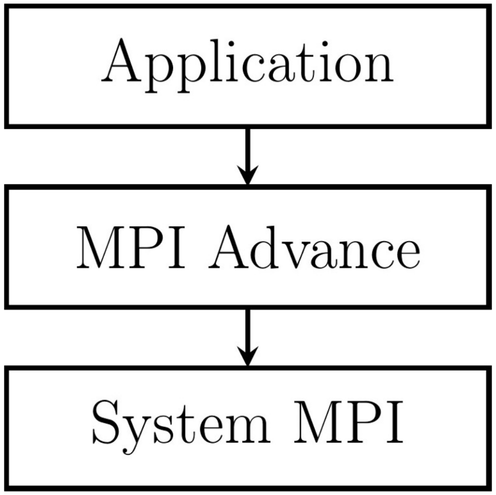

Application
MPI Advance
System MPI

Forthcoming Feature Set

- Partitioned Collective Communication APIs
  - superset of MPIPCL
  - pre-MPI-5 specification

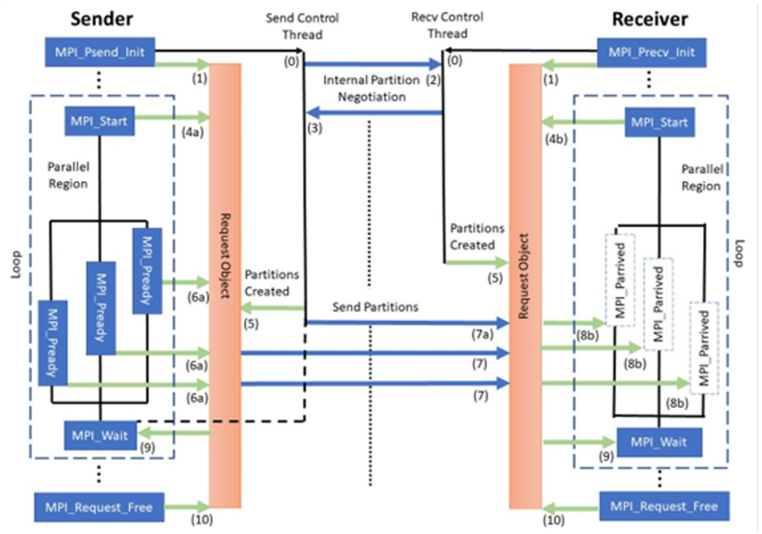
- A C++ interface for MPI-4.1 (based on MPL)

- Kokkos/MPI/C++ Integration/Wrappers (Sandia collaboration)



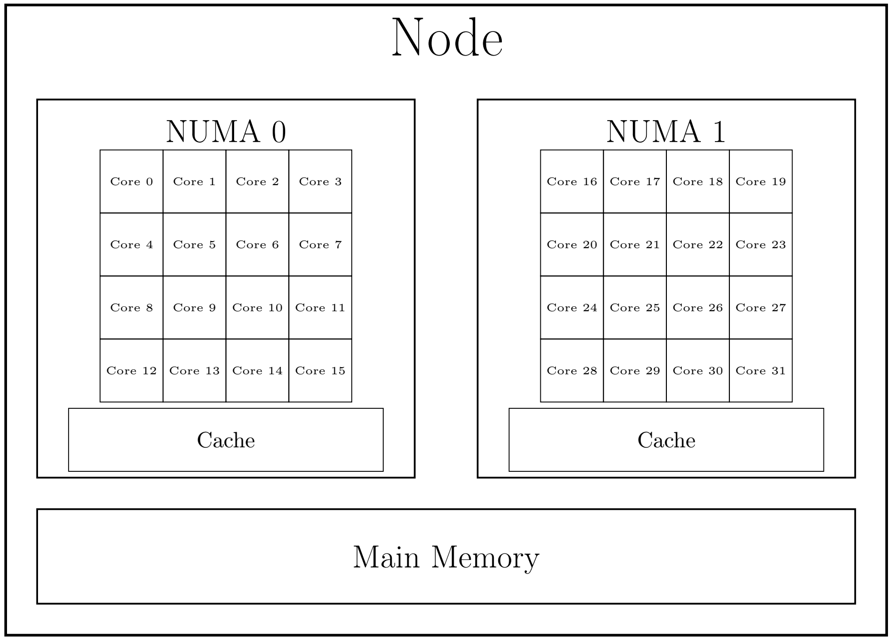
Partitioned Point-to-Point Operations (MPIPCL)

- Implements all MPI 4.0 partitioned communication APIs
- Layered library on top of existing MPI implementations
- Technical details:
  - Uses MPI Persistent P2P APIs
  - Uses progress thread for partition negotiation
- Partitioned collectives under prototyping and experimentation



M. G.F. Dosanjh, A. Worley, D. Schafer, P. Soundararajan, S. Ghafoor, A. Skjellum, P. V. Bangalore, R. E. Grant, Implementation and evaluation of MPI 4.0 partitioned communication libraries, *Parallel Computing*, Volume 108, 2021, https://doi.org/10.1016/j.parco.2021.102827.

Locality Aware MPI Collective Operations



Node

NUMA 0

Core 0	Core 1	Core 2	Core 3
Core 4	Core 5	Core 6	Core 7
Core 8	Core 9	Core 10	Core 11
Core 12	Core 13	Core 14	Core 15

Cache

NUMA 1

Core 16	Core 17	Core 18	Core 19
Core 20	Core 21	Core 22	Core 23
Core 24	Core 25	Core 26	Core 27
Core 28	Core 29	Core 30	Core 31

Cache

Main Memory

Example symmetric multiprocessor (SMP) node

Locality Region Examples:

- On-socket
- On-node
- On-node-group
- Proximity in topology

Locality Aware MPI Collective Operations

- Designed to fit codebases with minimal changes to existing code
- Locality-Aware Collectives: Allgather, Alltoall, Alltoallv
- Persistent Neighborhood Collectives:
  - Neighbor Alltoallv, Neighbor Alltoallw
  - Requires use of special topology communicator (dist_graph_create)
- Optimizations of the two above libraries w/ respect to GPUs
- Integrated into Hypre, Trilinos, Beatnik

Bienz A, Gropp WD, Olson LN. Reducing communication in algebraic multigrid with multi-step node aware communication. *The International Journal of High Performance Computing Applications.* 2020;34(5):547-561.

Break-out Session (Room: Arizona South)

- Session 1 (2 – 3pm) – Partitioned Point-to-Point Operations
  - Overview of partitioned point-to-point operations
  - Writing programs using partitioned point-to-point operations
  - Using MPIPCL
- Session 2 (3 – 4pm) – Locality Aware MPI Collective Operations
  - Overview of locality aware collective operations
  - Writing programs using locality aware collective operations
  - Using locality aware collective library

Questions?

Funding in part is acknowledged from the U.S. Department of Energy's National Nuclear Security Administration (NNSA) under the Predictive Science Academic Alliance Program (PSAAP-III), Award DE-NA0003966.