A Portable Implementation of Partitioned Point-to-Point Communication Primitives

by
Purushotham Bangalore, Andrew Worley, Derek Schafer
Ryan Grant, Anthony Skjellum, Sheikh Ghafoor

EuroMPI 2020 - 21-24 September 2020 Austin, TX

Tennessee TECH

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

Sandia National Laboratories

THE UNIVERSITY OF TENNESSEE AT CHATTANOOGA

Partitioned Communication

- Recent addition to MPI 4.0 Standard
- Designed to interface with hybrid programming practices
- Allows a buffer to be broken into sections with each section being sent as soon as it is ready
- High optimization potential
- Few (if any) public implementations

Overview

What we are introducing today

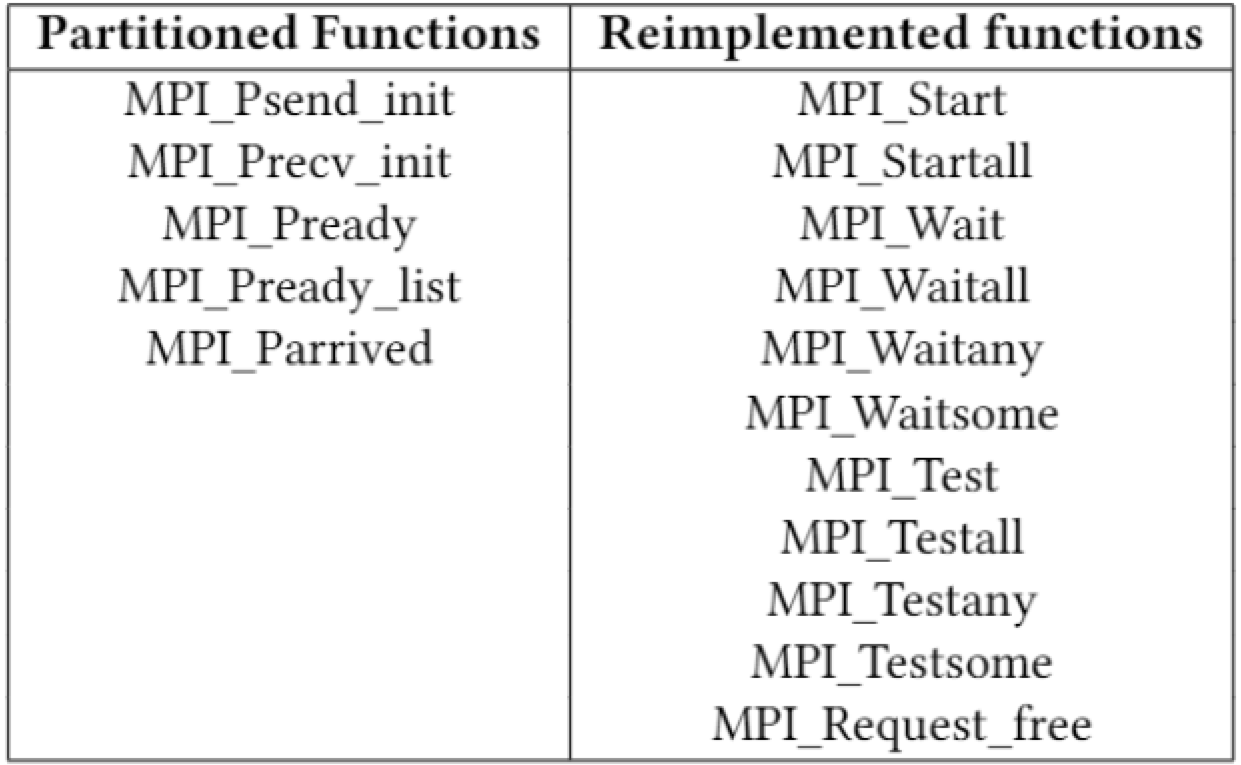
- Partitioned Communication Library (MPIPCL)
- A portable library that implements the partitioned communication API.
- Utilizes the Point to Point communications of the host

Why did we make this?

- Increase exposure to the new communication model
- Allow observation of the model's behavior
- Act as a prototype for future integrations/implementations

Library API

- Implements approved API
- Added MPIX versions to interact with the new request object
- Do not return valid MPI_Status objects

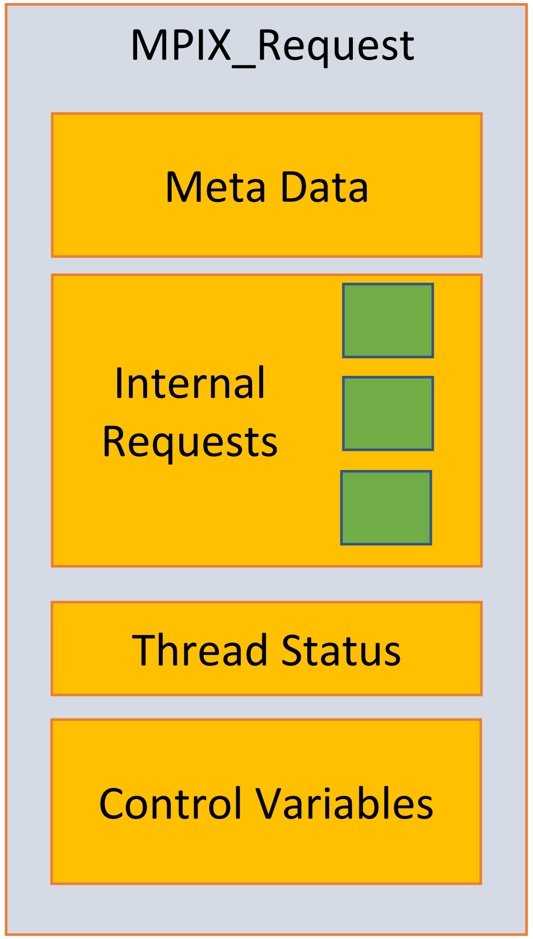


Partitioned Functions	Reimplemented functions
MPI_Psend_init	MPI_Start
MPI_Precv_init	MPI_Startall
MPI_Pready	MPI_Wait
MPI_Pready_list	MPI_Waitall
MPI_Parrived	MPI_Waitany
	MPI_Waitsome
	MPI_Test
	MPI_Testall
	MPI_Testany
	MPI_Testsome
	MPI_Request_free

Library Control Object– MPIX_Request

- MPI_Requests have no way to track partial completion

- MPIX_Request
  - Wrapper around several internal requests
  - Contains tracking & metadata on the overall request



Basic Execution

Sender

MPI_Psend_Init()

MPI_Start()

...*Prepare partition(s)*...

MPI_Pready()

…

MPI_Wait()

MPI_Request_free()

Receiver

MPI_Precv_Init()

MPI_Start()

...

...*Wait for partition(s)*…

MPI_Parrived()

MPI_Wait()

MPI_Request_free()

Internal Behavior – Partition Abstraction

Benefits

- Allows the library to use a different number of partitions on each side
- Adds multiple internal optimization options

Challenges:

- How to map internal requests to external partitions
  - Basic offset mapping
- Point-to-Point requires matching send and receive calls.
  - How to communicate number of internal requests

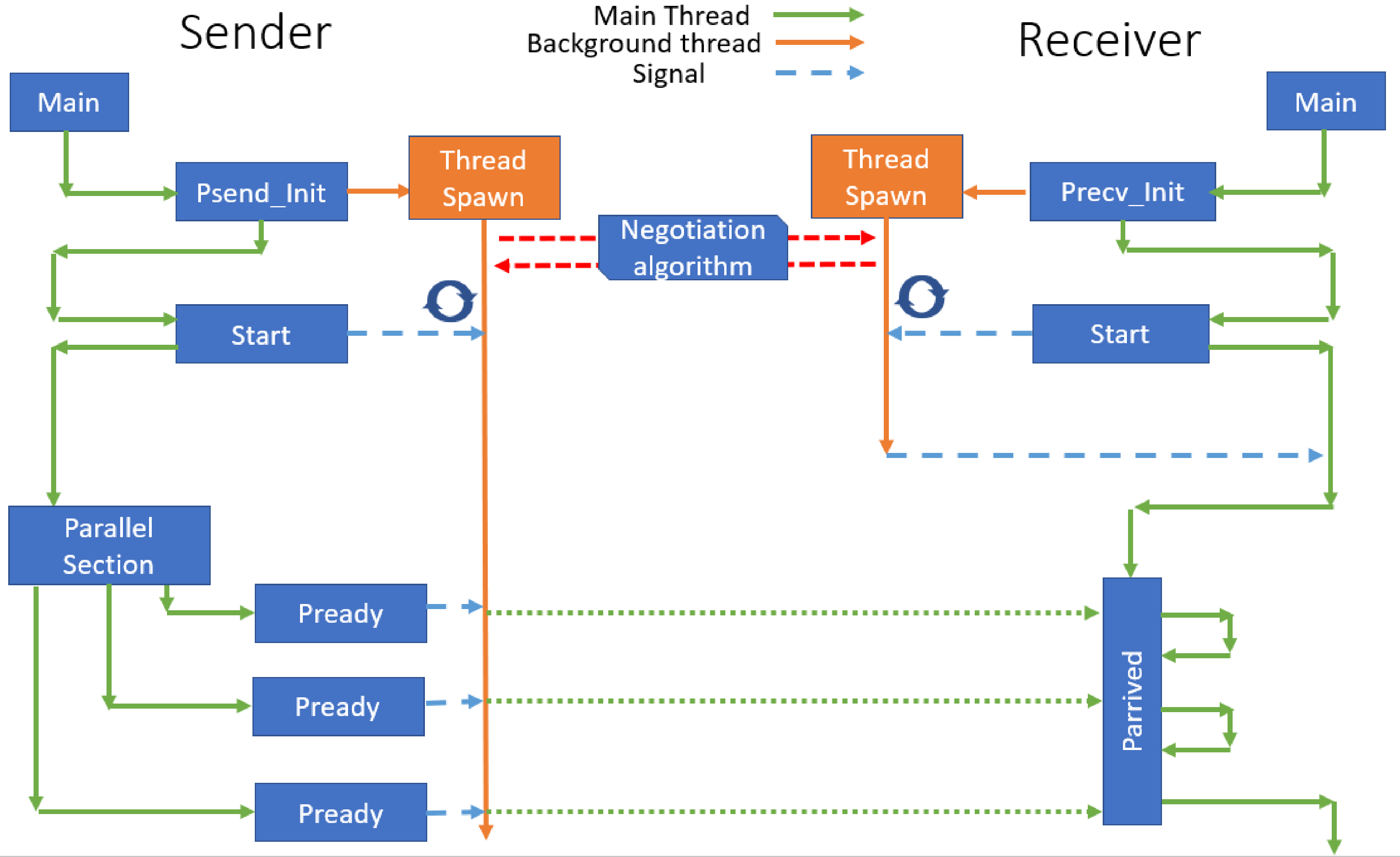
Internal Behavior – Partition Abstraction

Synchronization Solutions:

- Modify the communicator
  - the library does not have access the internal calls

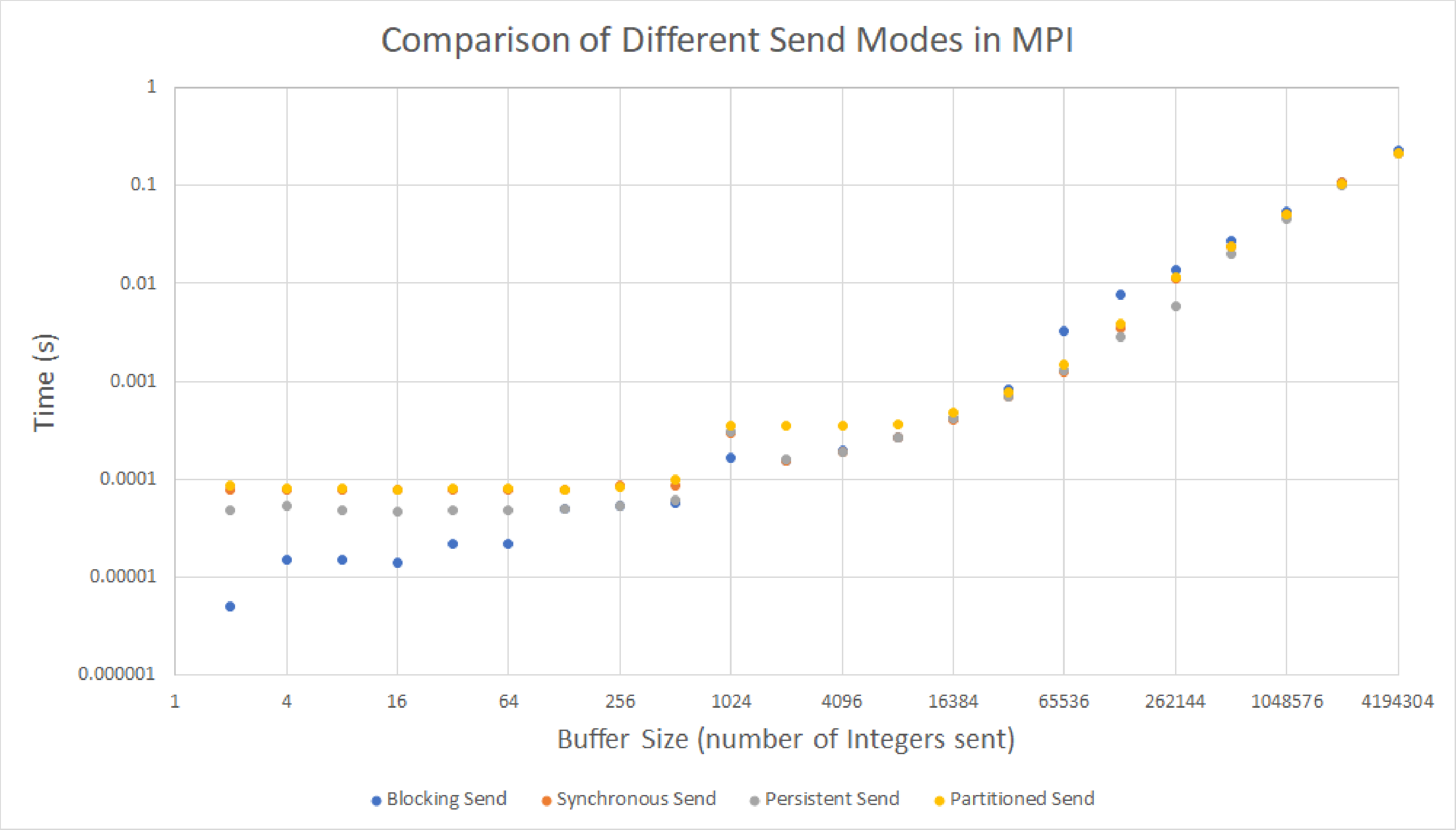
- Have processes communicate directly
  - Point-to-Point communication is blocking on receive.
    - Have a thread block in the background

Internal Control Flow



Performance Benchmarking

- Comparison with existing send modes
- Increasing Buffer Size
- Single Partition
- Small overhead, but growth is as expected



Limitations and Future Work

- Limitations
  - Only supports contiguous datatypes
  - Does not return MPI_Status Objects
  - Limited number of negotiation algorithms
- Future Work
  - Expand datatype support
  - Add custom status object
  - Integrate with Psync (once finalized)
  - Additional optimizations internally

Questions?