MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

MV2-FPGA
Bringing MPI Support to FPGAs

Nicholas Contini

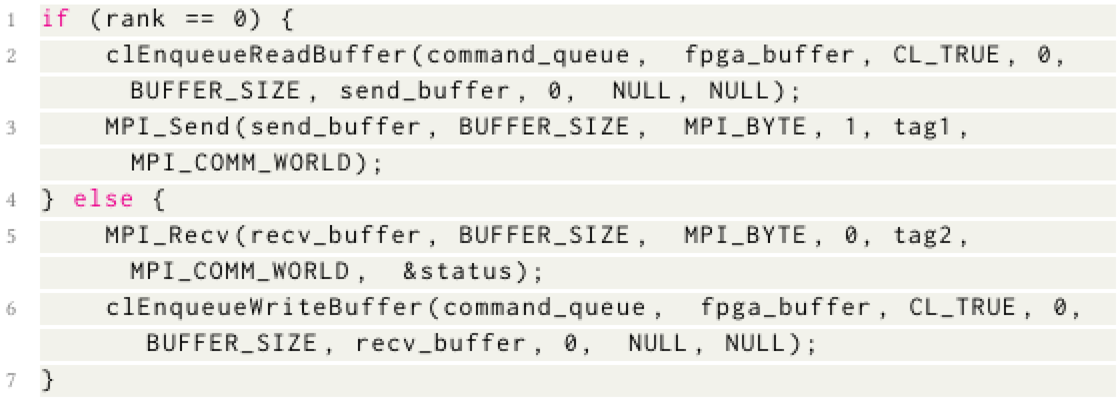
Department of Computer Science and Engineering
The Ohio State University

Motivation

- Interest in Reconfigurable HPC is growing
  - Investment in HLS by major FPGA vendors has increased programmability
  - Less need for HDL
- New familiar interfaces like OpenCL, C/C++, and Python
- Despite this, integrating FPGAs into applications, both old and new still remains a challenge.
  - E.g. lack of support for inter-FPGA communication within MPI implementations.

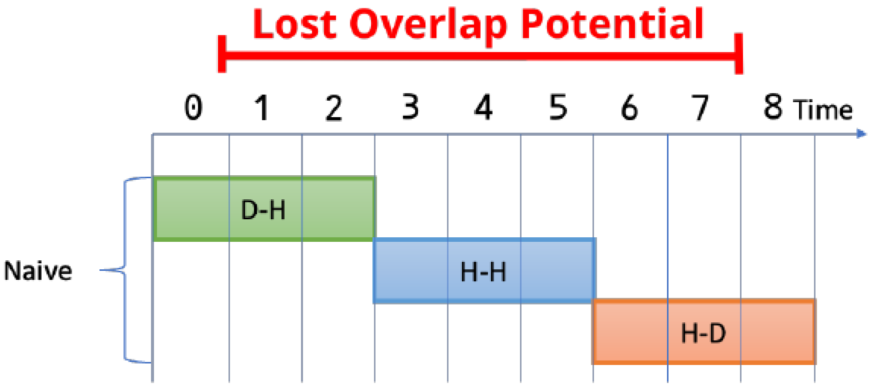
Motivation

- Why is MPI support so important?
  - Removes code from the application level
  - Enables optimizations that are otherwise not possible
  - MPI developers integrate new vendor features, application writers reap the benefits.



```
if (rank == 0) {
    clEnqueueReadBuffer(command_queue, fpga_buffer, CL_TRUE, 0,
      BUFFER_SIZE, send_buffer, 0, NULL, NULL);
    MPI_Send(send_buffer, BUFFER_SIZE, MPI_BYTE, 1, tag1,
      MPI_COMM_WORLD);
} else {
    MPI_Recv(recv_buffer, BUFFER_SIZE, MPI_BYTE, 0, tag2,
      MPI_COMM_WORLD, &status);
    clEnqueueWriteBuffer(command_queue, fpga_buffer, CL_TRUE, 0,
       BUFFER_SIZE, recv_buffer, 0, NULL, NULL);
}
```

A naive implementation of MPI-based inter-FPGA communication without MPI-level FPGA support

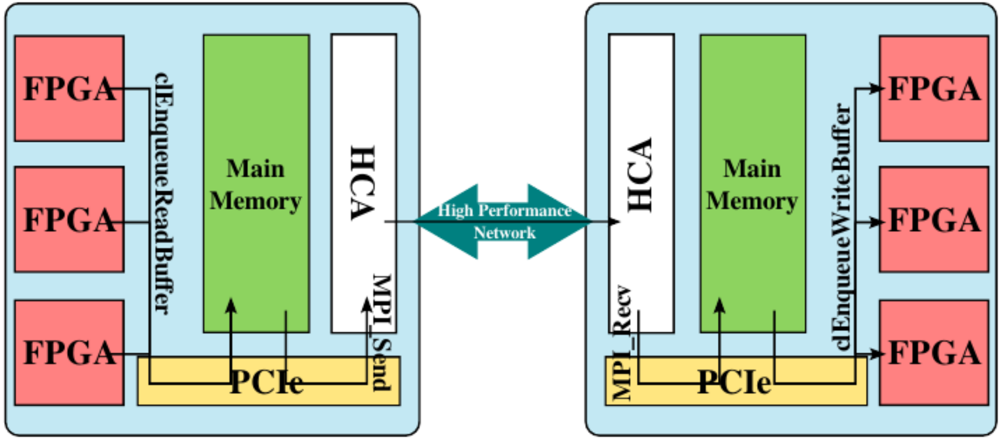


Problem Statement

- Can support for inter-FPGA communication be integrated at the MPI-level?
- How can we make this accessible to mainstream users?
- Can we optimize data movement between FPGAs, internode and intranode?

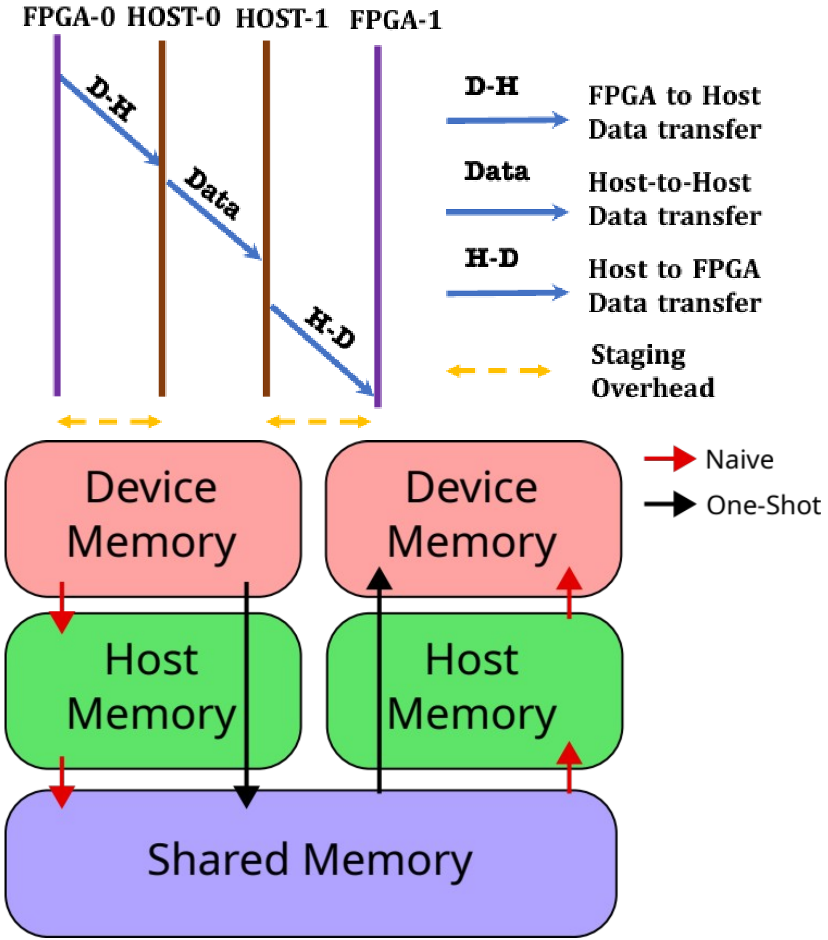
Basic FPGA Support

- OpenCL utilized by many FPGA applications
- Track buffers by “capturing” clCreateBuffer
- Appropriate handling can be done within MPI when these buffers are identified in the runtime



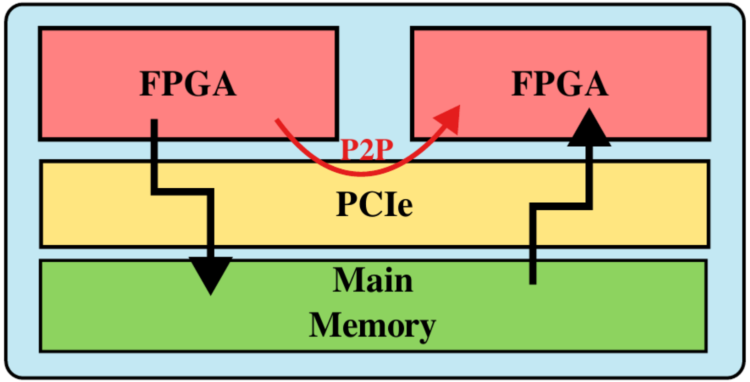
One-Shot Design (Intranode)

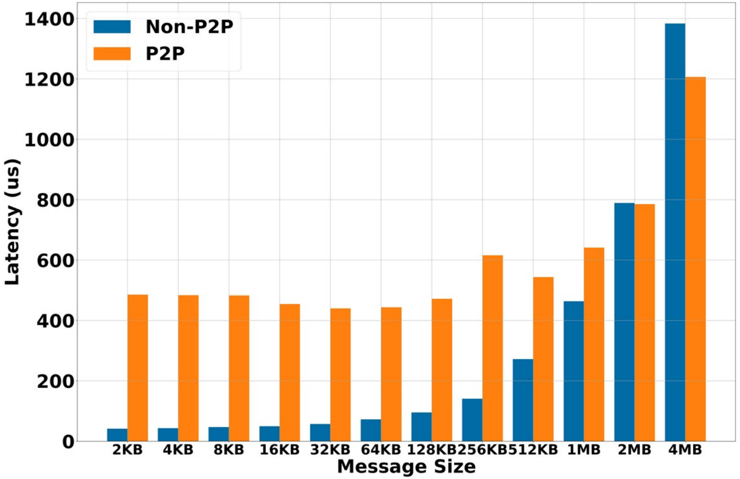
- For small message sizes, an “eager” protocol is used
  - Very little coordination between sender and receiver
  - Sender quickly places data into shared memory and does not wait for data to be consumed
  - If receiver moves data into application buffer whenever its ready to receive.
- Compared to the naive approach to inter-FPGA MPI communication, this avoids two copies



P2P Design (Intranode)

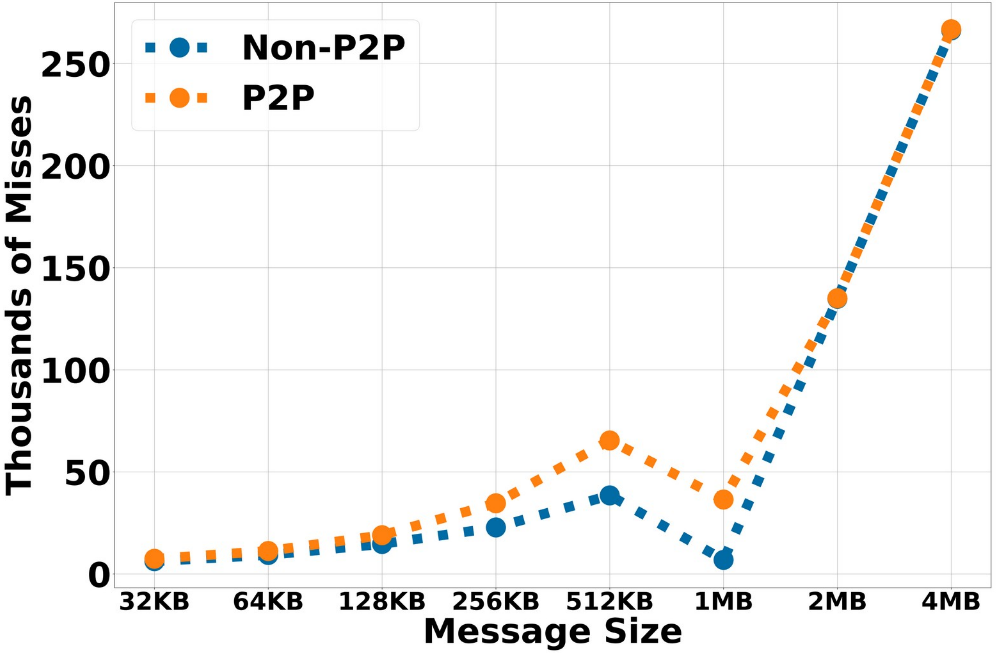
- What about large messages?
- P2P transfers enable data to travel between devices over PCIe without entering host memory
- However, P2P transfers aren’t always better than baseline





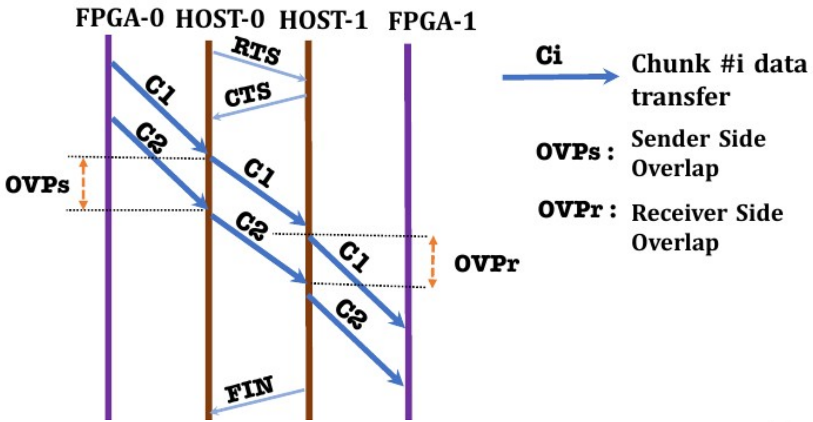
P2P Design (Intranode)

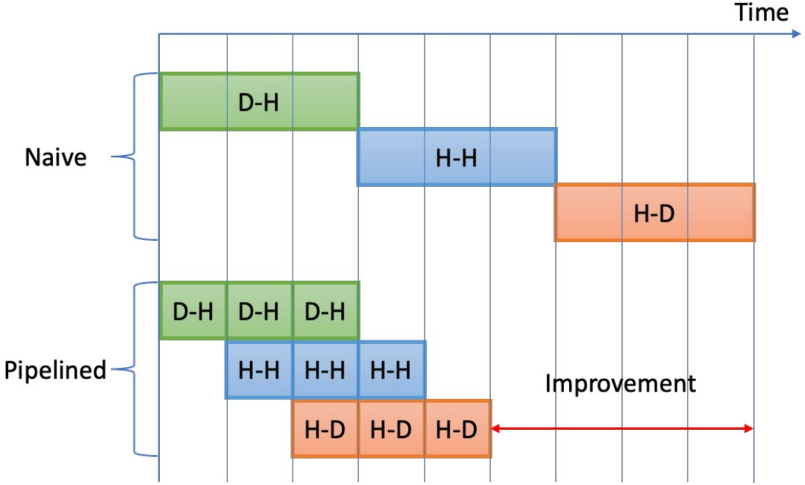
- Performance degradation appears linked to cache performance
- This renders this protocol useful at 2MB and above



Pipeline Design (Internode)

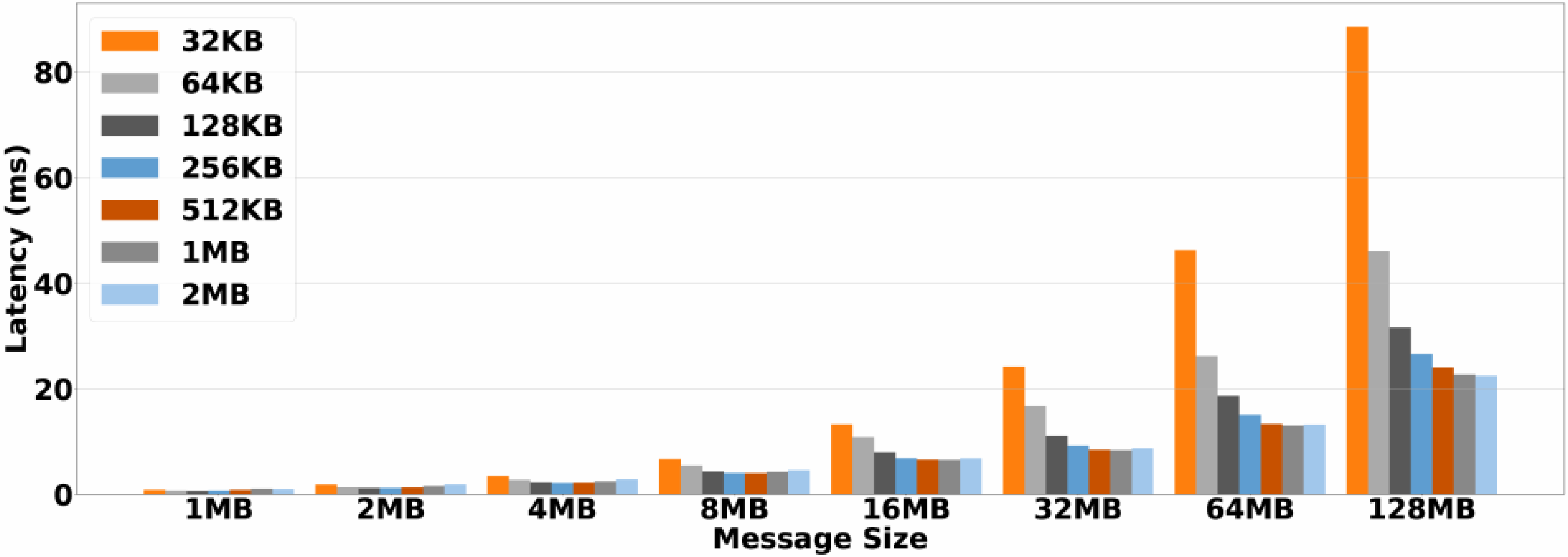
- Staging costs dominate small message operations
- At larger sizes network latency start to match staging latency
- This gives us an opportunity to pipeline different stages of communication





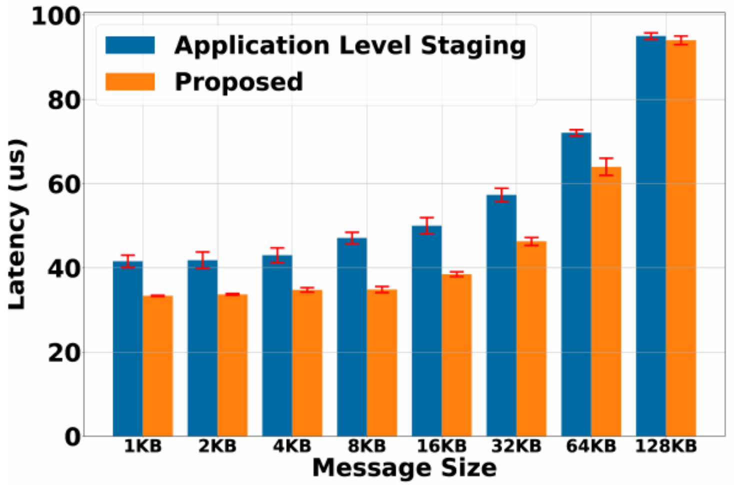
Pipeline Design (Internode)

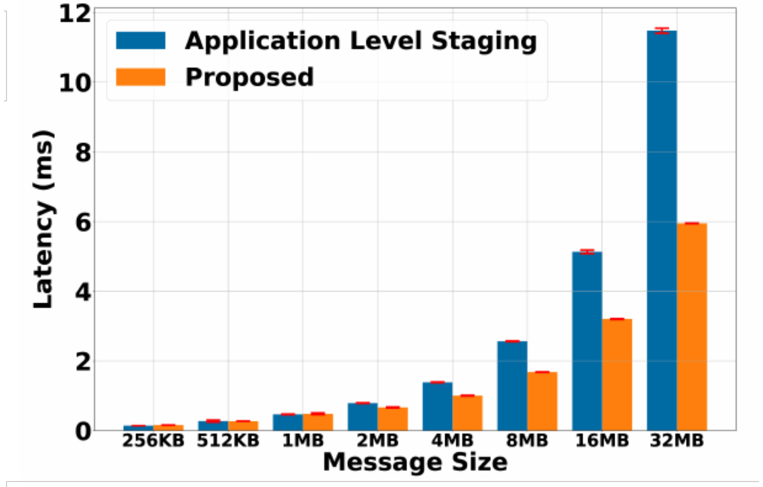
Testing latency of pipeline design with different chunk sizes



Results

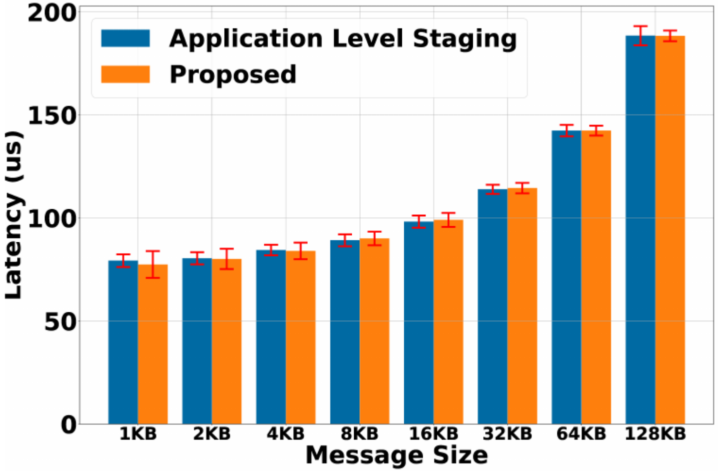
- One Shot used up to 256K
  - Up to 25% improvement
- P2P used starting at 2MB
  - Up to 45% improvement

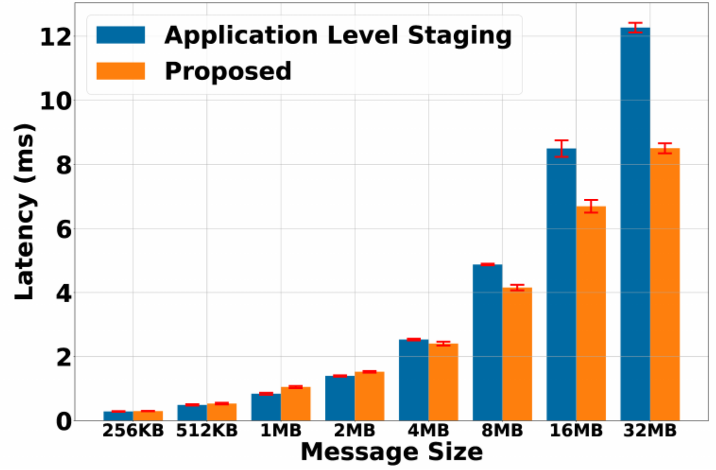




Results

- Basic support used for lower message sizes
  - Matches baseline performance
- Pipelining used starting at 2MB
  - Up to 33% improvement





Conclusion

- Providing FPGA-support within MPI can increase productivity and performance
- Some optimizations are not even possible without this support
- Future enhancements to MVAPICH can benefit applications already using FPGA-support for "free" by simply installing newer versions
- Future Work
  - Utilize our designs in FPGA-based applications to enable scaling-out
  - Explore ways to use dedicated FPGA networks within MPI

For More Info See

- Contini, N., Ramesh, B., Kandadi Suresh, K., Tran, T., Michalowicz, B., Abduljabbar, M., Subramoni, H., & Panda, D. (2023, June). Enabling Reconfigurable HPC through MPI-based Inter-FPGA Communication. In *Proceedings of the 37th International Conference on Supercomputing* (pp. 477-487).

THANK YOU!

Network Based Computing Laboratory

Network-Based Computing Laboratory

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

MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

The High-Performance MPI/PGAS Project

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

HiBD
High-Performance
Big Data

The High-Performance Big Data Project

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

HiDL
High-Performance
Deep Learning

The High-Performance Deep Learning Project

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