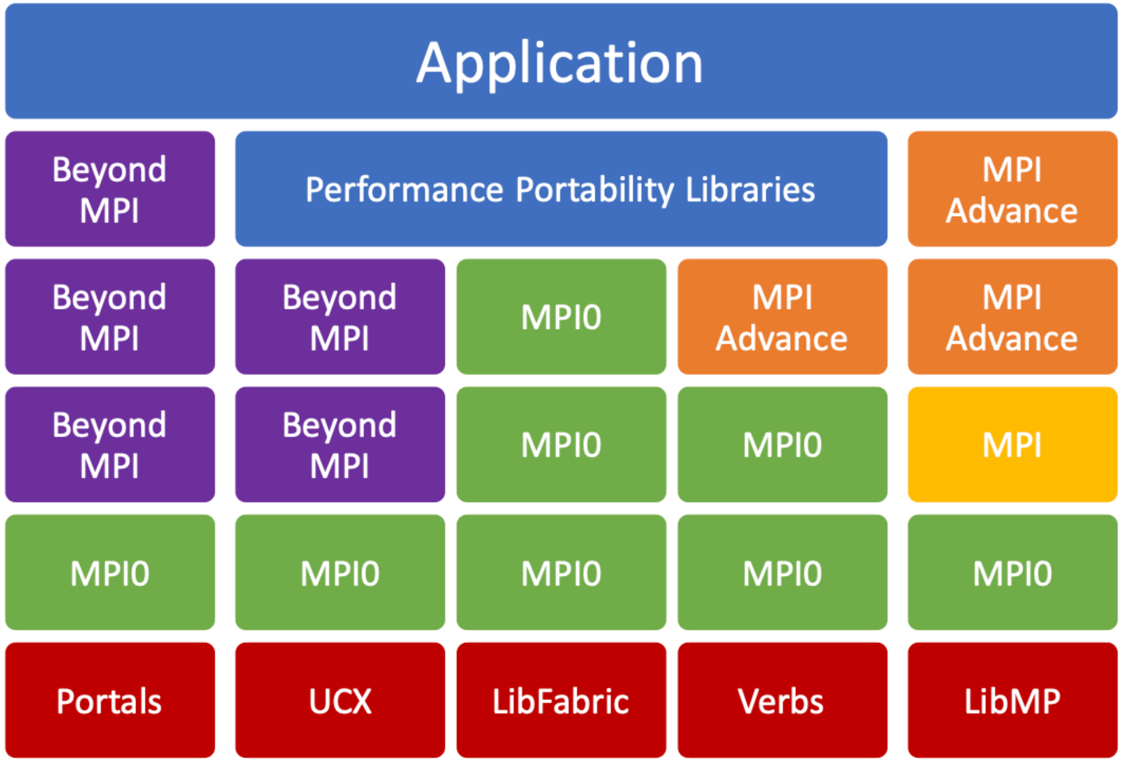
Irregular Communication

Amanda Bienz

Assistant Professor
Department of Computer Science
University of New Mexico

Talk Overview

- Portable optimizations for codes with irregular communication
  - MPI Advance in HYPRE and Trilinos
  - Optimization of HYPRE using neighbor collectives
  - Optimization of GPU-based all-to-all communication
  - Performance analysis of topology identification algorithms
  - Designing abstractions to improve topology identification and topology-based neighbor communication



Application

Beyond MPI | Performance Portability Libraries | MPI Advance

Beyond MPI | Beyond MPI | MPI0 | MPI Advance | MPI Advance

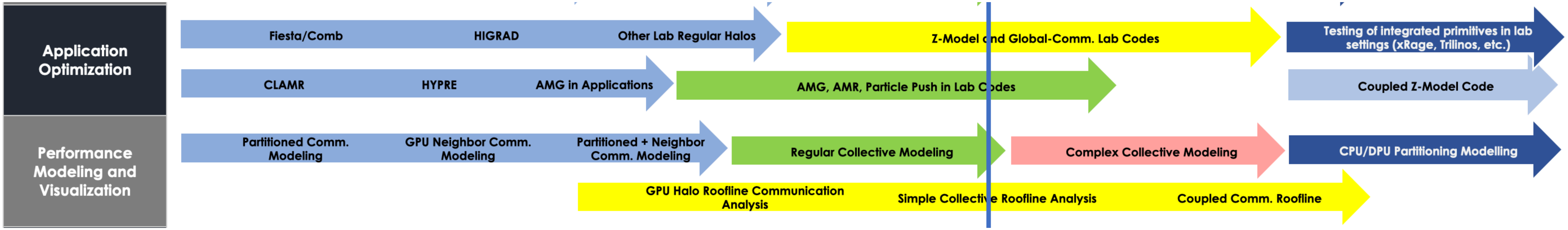
Beyond MPI | Beyond MPI | MPI0 | MPI0 | MPI

MPI0 | MPI0 | MPI0 | MPI0 | MPI0

Portals | UCX | LibFabric | Verbs | LibMP

Updated 5-year Project Roadmap

- Benchmarking and modeling for irregular and global communication
- Portable optimizations for lab codes that rely on irregular and global communication



Application Optimization

Fiesta/Comb

HIGRAD

Other Lab Regular Halos

Z-Model and Global-Comm. Lab Codes

Testing of integrated primitives in lab settings (xRage, Trilinos, etc.)

CLAMR

HYPRE

AMG in Applications

AMG, AMR, Particle Push in Lab Codes

Coupled Z-Model Code

Performance Modeling and Visualization

Partitioned Comm. Modeling

GPU Neighbor Comm. Modeling

Partitioned + Neighbor Comm. Modeling

Regular Collective Modeling

Complex Collective Modeling

CPU/DPU Partitioning Modelling

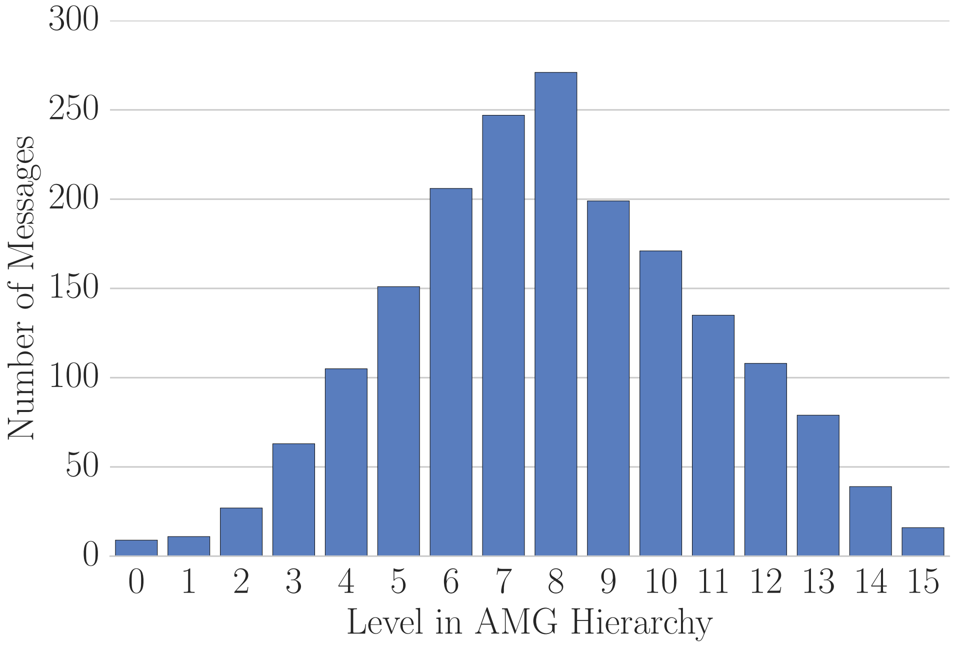
GPU Halo Roofline Communication Analysis

Simple Collective Roofline Analysis

Coupled Comm. Roofline

Motivation - Neighbor Collectives

- Communication is typically the bottleneck in irregular parallel applications
- Often, each application or solver will implement their own communication optimizations
    - Some really clever approaches! But no central knowledge, so people keep reinventing the wheel
- Many parallel codebases have existed for decades
    - Want to optimize performance with minimal changes to existing codebases



Approach

1. Profile systems and form representative performance models

2. Use performance models to create communication optimizations

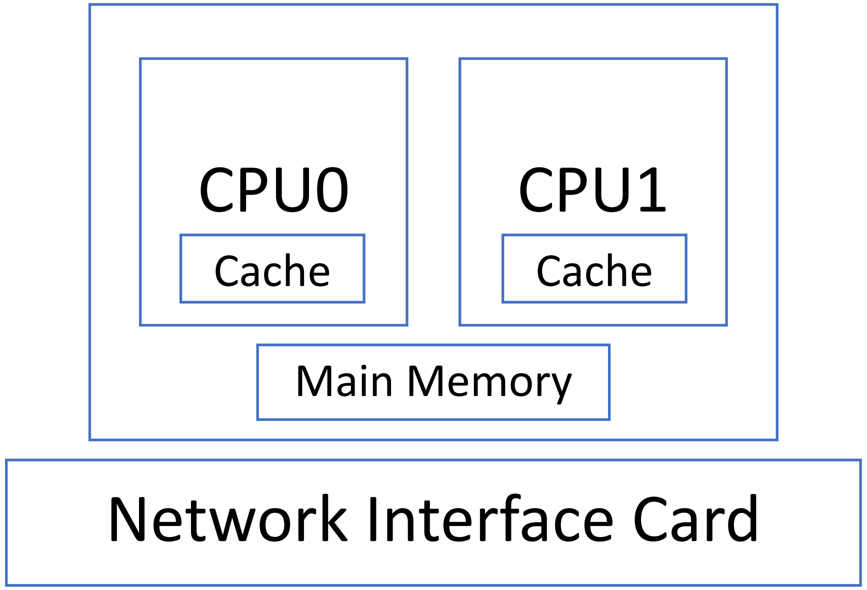
3. Add optimizations to MPI Advance to improve performance of existing applications

MPI Advance

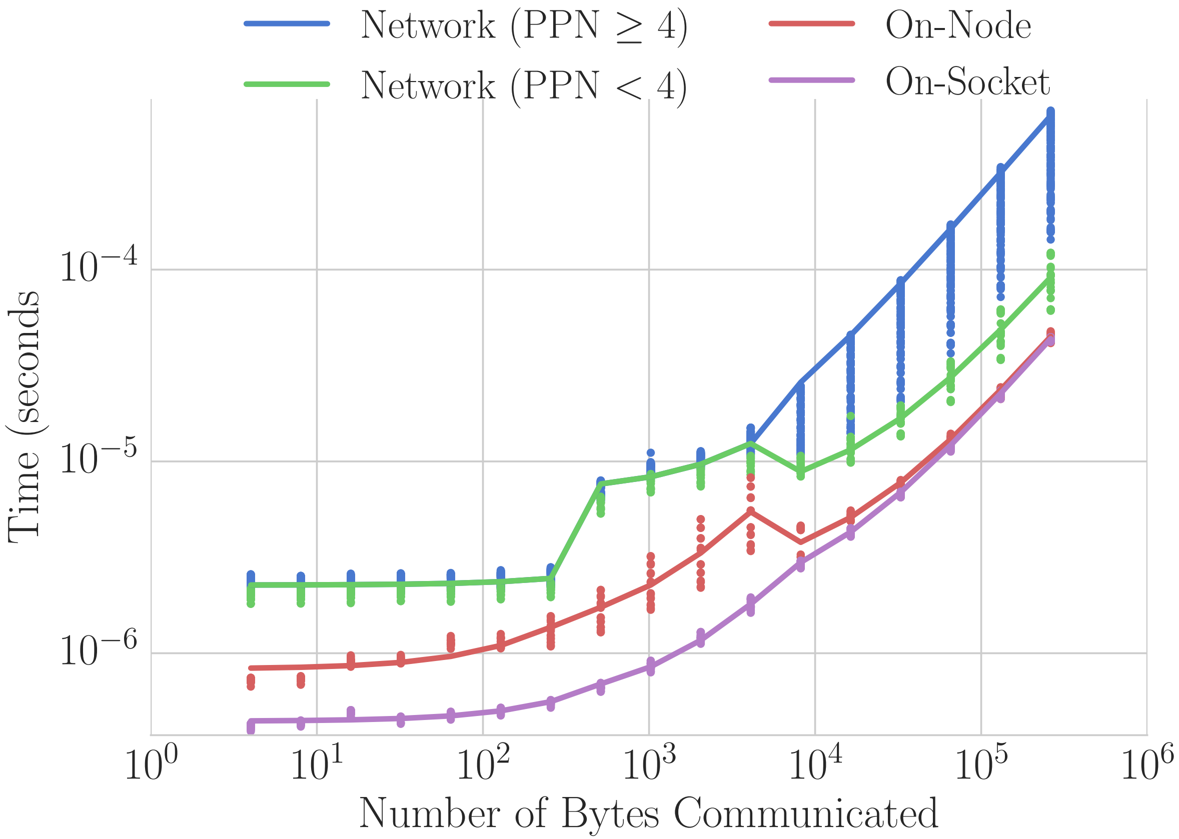
- Lightweight library, sits on top of MPI
  - Utilizes underlying communication of system MPI installation
- All optimizations covered in this talk have been added to MPI Advance, allowing for others to use these optimizations through the MPIX extension.
- **GPU-Aware support**

- MPI Advance: Open-Source Message Passing Optimizations (https://eurompi23.github.io/assets/papers/EuroMPI23_paper_33.pdf)

Symmetric Multiprocessing Architectures

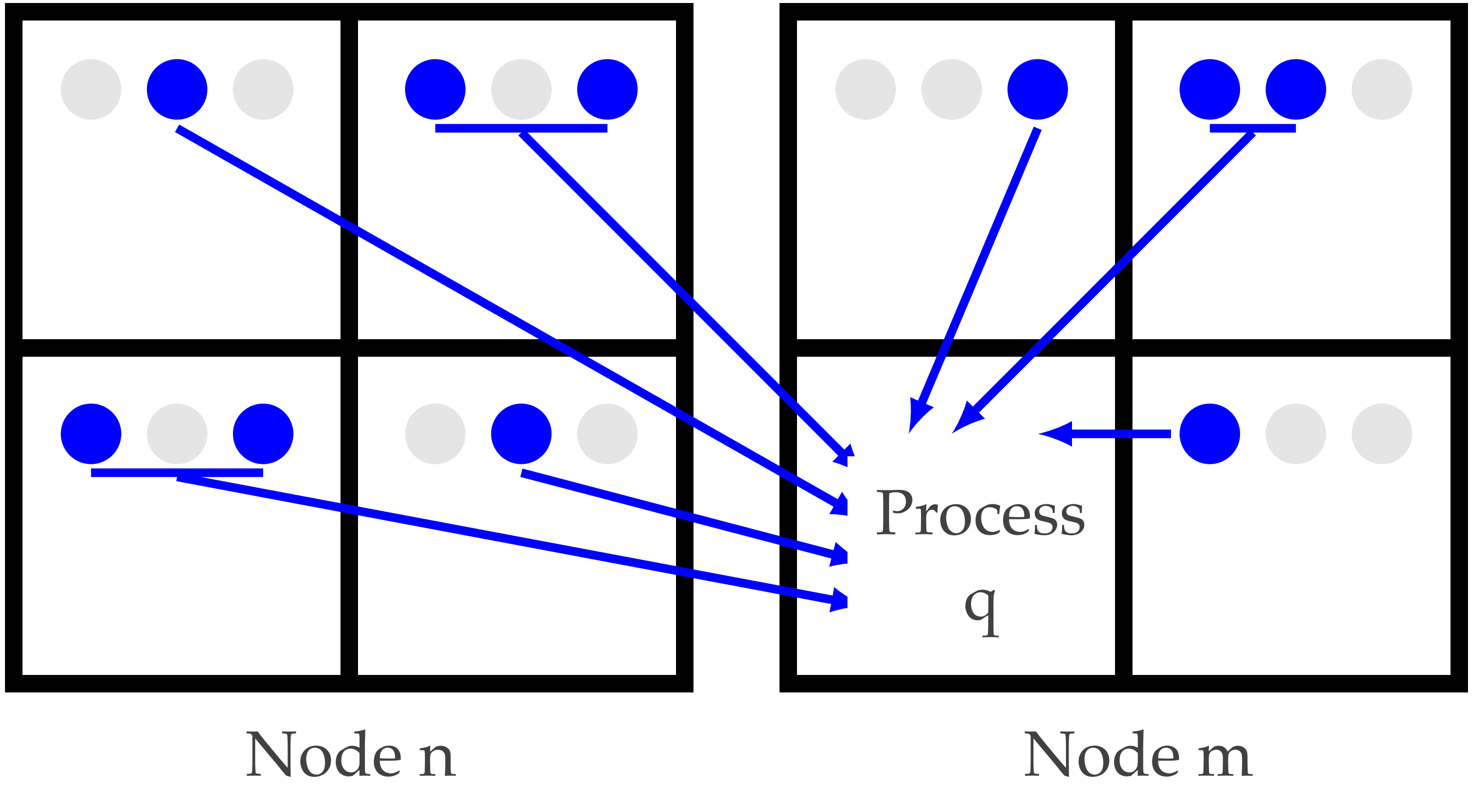


Symmetric Multiprocessing (SMP) Node



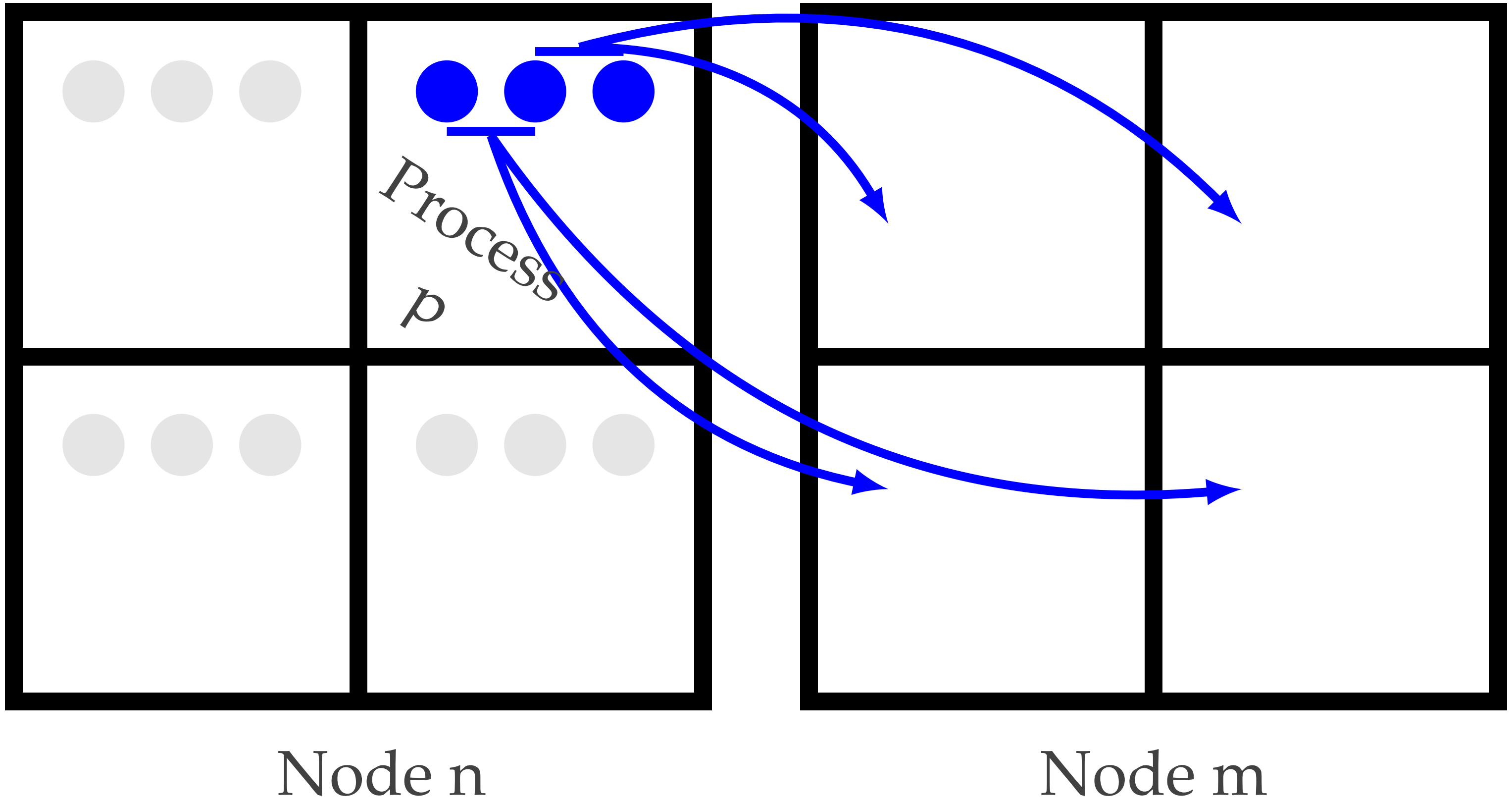
Key Takeaway : Intra-socket << Intra-node/Inter-socket << Inter-node

Standard Communication



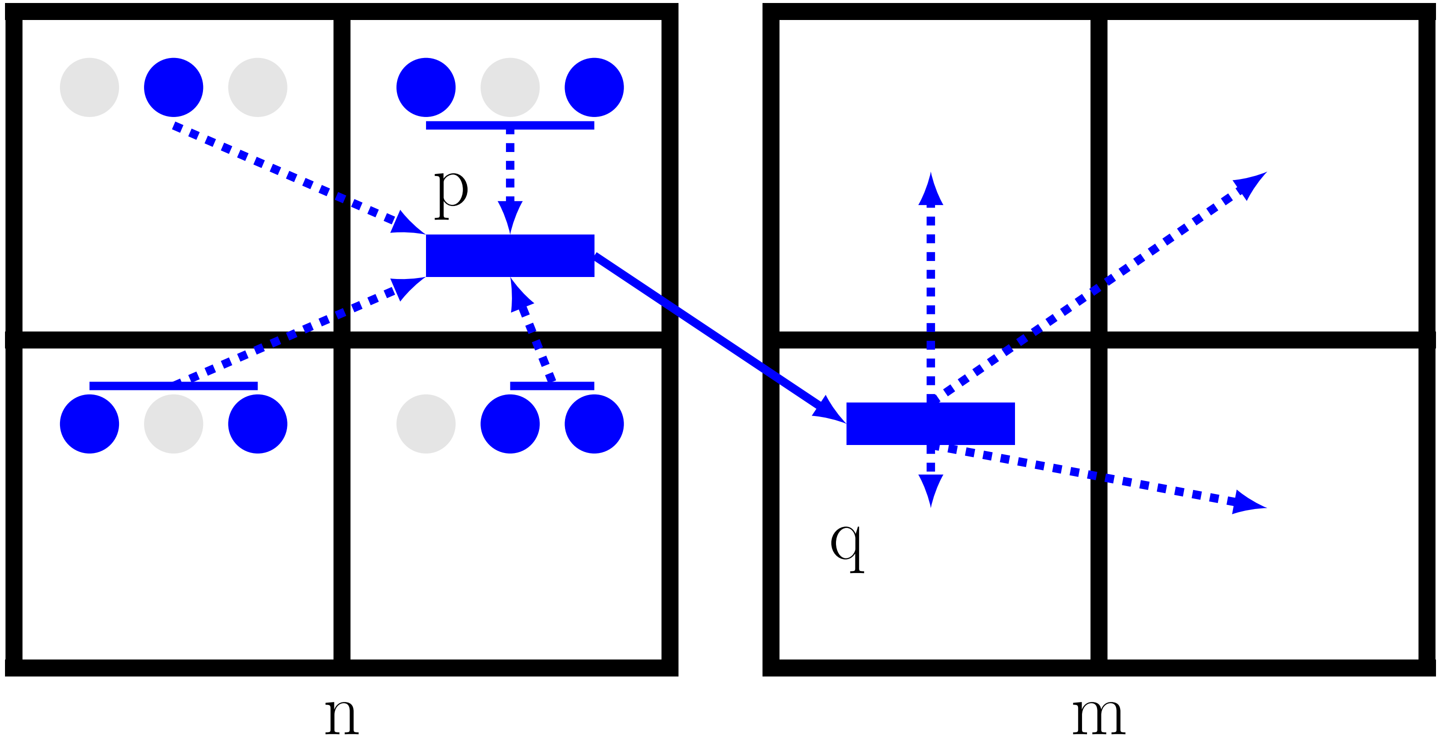
Multiple messages between set of nodes

Standard Communication



Multiple messages and duplicate data between set of nodes

Locality-Aware Communication



All processes per node are active in communication

Irregular Communication Steps

Point-to-Point Communication :

1. Form communication package

2. MPI_Send_init(s)

3. MPI_Recv_init(s)

4. Iterative MPI_Startall/MPI_Waitall

Irregular Communication Steps

Point-to-Point Communication :

1. Form communication package
2. MPI_Send_init(s)
3. MPI_Recv_init(s)
4. Iterative MPI_Startall/MPI_Waitall

Neighbor Collective :

1. Form communication package
2. MPI_Dist_graph_create_adjacent
3. MPI_Neighbor_alltoallv_init
4. Iterative MPI_Start/MPI_Wait

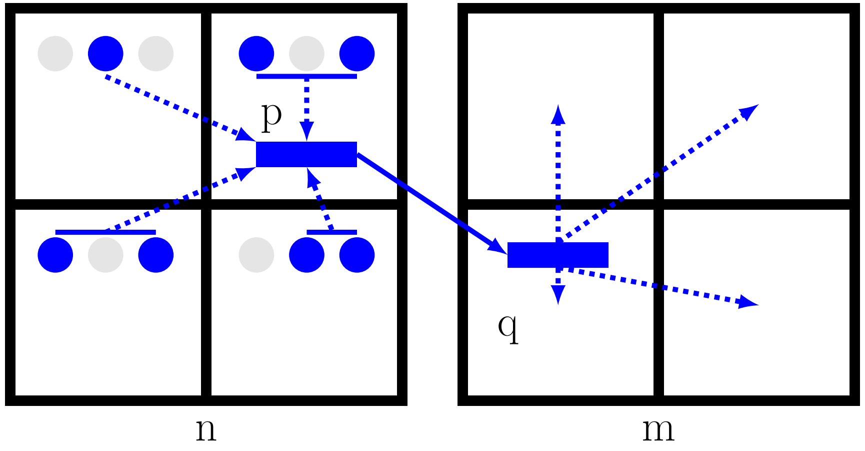
Neighbor Collectives in HYPRE

- Solvers such as HYPRE each implement irregular communication (e.g. Isends/Irecvs)

- Gerald Collom has spent two summers working with the HYPRE team at LLNL
  - Integrated and analyzed MPI Advance locality-aware neighborhood collectives within the solve phase of HYPRE

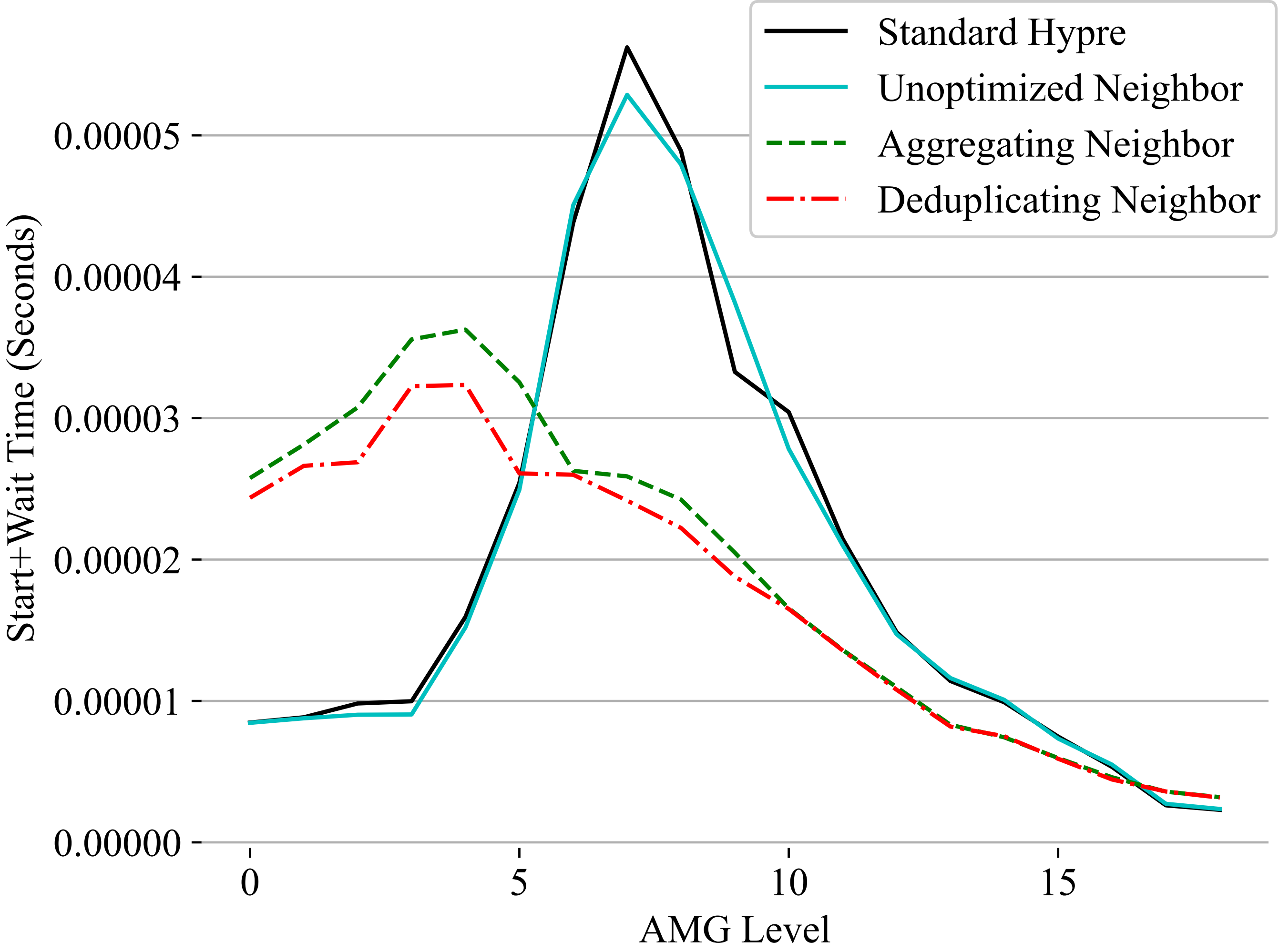
- Paper accepted to ExaMPI at SC23

MPI Advance Neighbor Collectives



- Persistent : allows setup costs associated with optimizations to occur once in MPI_Neighbor_alltoallv_init
- Unoptimized : wraps standard communication
- Aggregated : concatenates all messages
- De-duplicate : extended interface, only sends each index between set of regions one time

Neighbor Collectives in Hypre



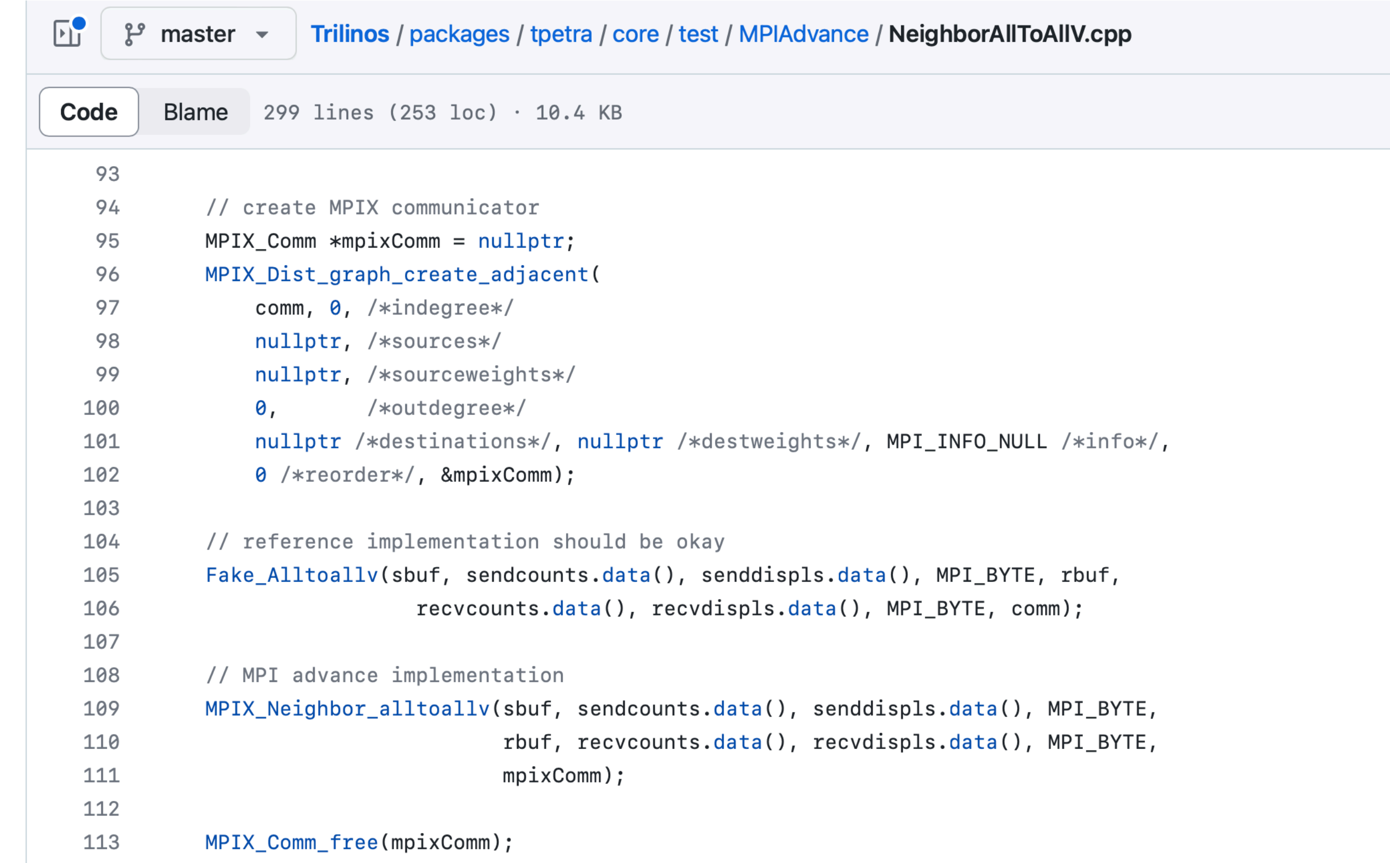
- Per-iteration costs greatly reduced on coarse levels

- Gerald’s poster

- Optimizing Irregular Communication with Neighborhood Collectives and Locality-Aware Parallelism (https://arxiv.org/pdf/2306.01876.pdf)

Integrated within Trilinos

- Mike Adams added MPI Advance into Trilinos during summer internship


master
Trilinos / packages / tpetra / core / test / MPIAdvance / NeighborAllToAllV.cpp
Code
Blame
299 lines (253 loc) · 10.4 KB

```

// create MPIX communicator
MPIX_Comm *mpixComm = nullptr;
MPIX_Dist_graph_create_adjacent(
    comm, 0, /*indegree*/
    nullptr, /*sources*/
    nullptr, /*sourceweights*/
    0,       /*outdegree*/
    nullptr /*destinations*/, nullptr /*destweights*/, MPI_INFO_NULL /*info*/,
    0 /*reorder*/, &mpixComm);

// reference implementation should be okay
Fake_Alltoallv(sbuf, sendcounts.data(), senddispls.data(), MPI_BYTE, rbuf,
                 recvcounts.data(), recvdispls.data(), MPI_BYTE, comm);

// MPI advance implementation
MPIX_Neighbor_alltoallv(sbuf, sendcounts.data(), senddispls.data(), MPI_BYTE,
                        rbuf, recvcounts.data(), recvdispls.data(), MPI_BYTE,
                        mpixComm);

MPIX_Comm_free(mpixComm);
```

Irregular Communication Steps

Point-to-Point Communication :

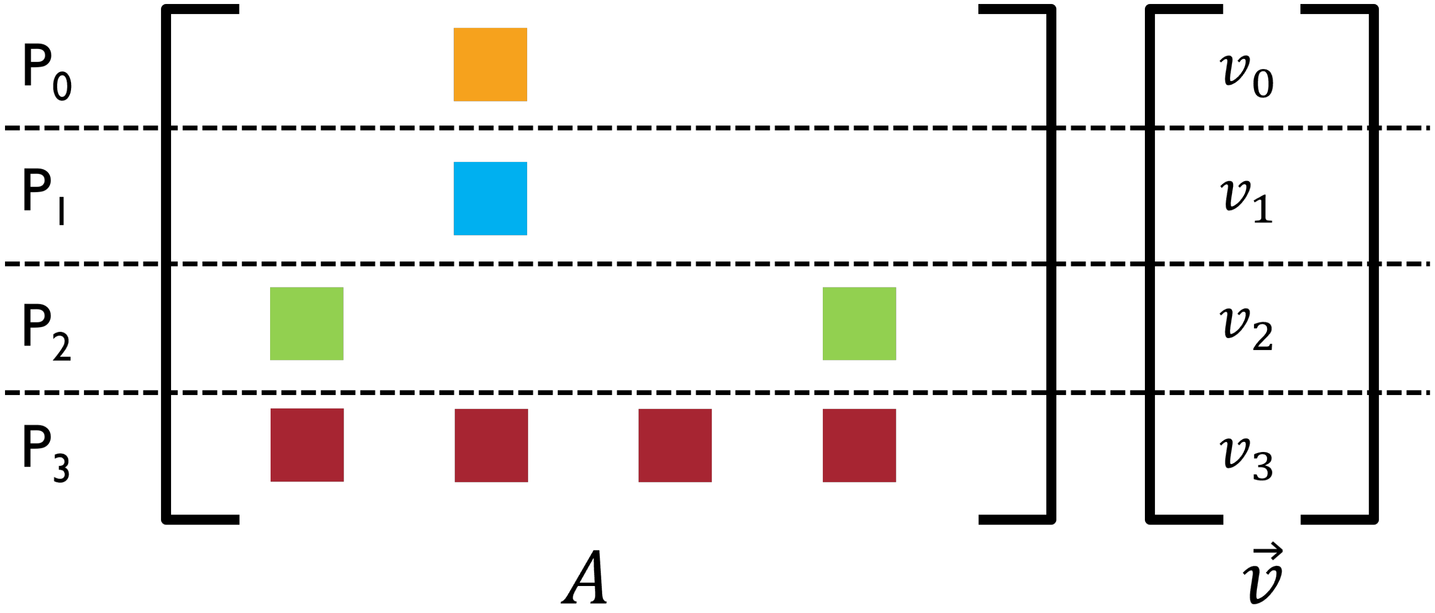
1. Form communication package
2. MPI_Send_init(s)
3. MPI_Recv_init(s)
4. Iterative MPI_Startall/MPI_Waitall

Neighbor Collective :

1. Form communication package
2. MPI_Dist_graph_create_adjacent
3. MPI_Neighbor_alltoallv_init
4. Iterative MPI_Start/MPI_Wait

Form Communication Pattern

- Receive side : fully local
- Send side :
  - Difficult
  - Requires dynamic communication
  - Unexpected messages



Form Communication Pattern

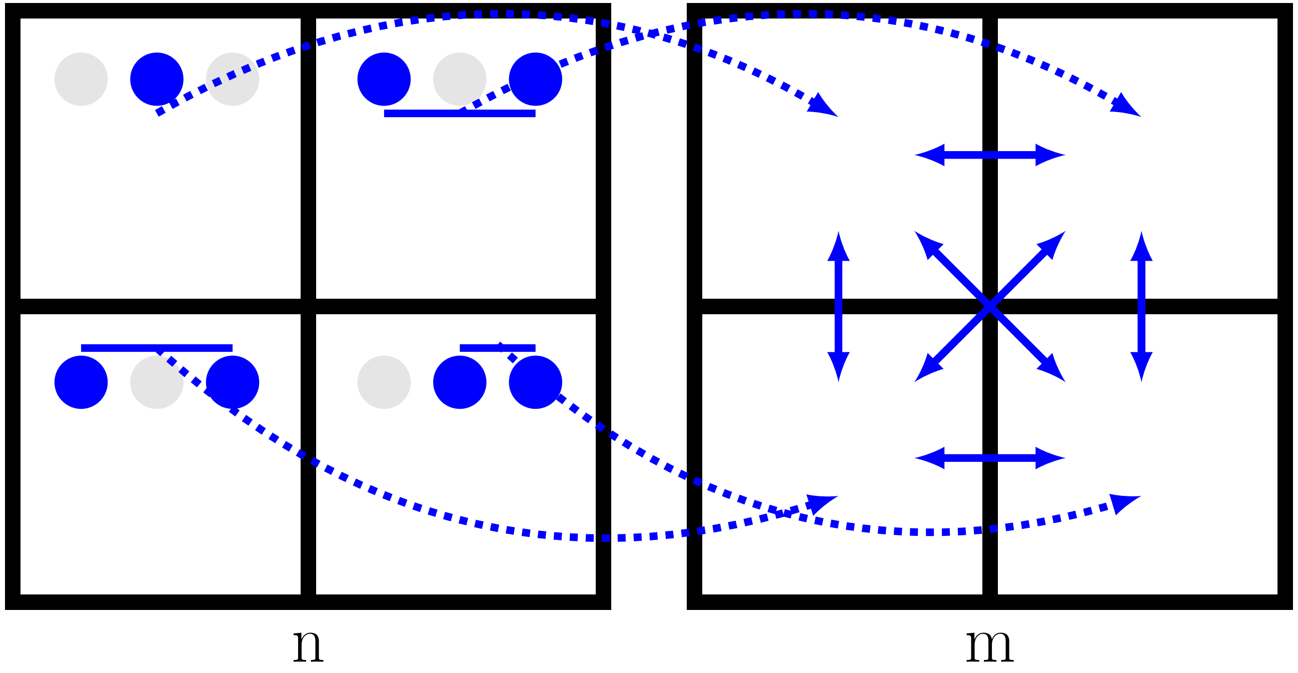
- Existing approaches :

  1. Allreduce to find how much data to receive, probe until you have received all
  2. Use synchronous sends and non-blocking probes to receive messages until all processes have completed all sends

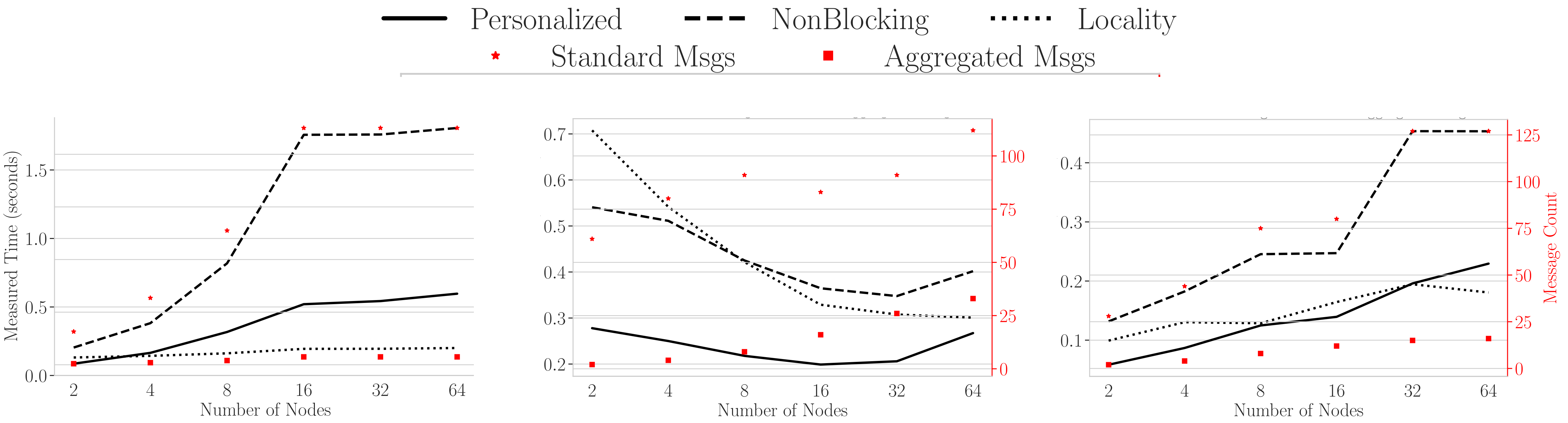
- Andrew Geyko has been analyzing these methods for bottlenecks
  - Initial hypothesis : can improve performance using RMA to avoid unexpected messages
  - Actual performance : dynamic receives *are actually cheaper* than standard point-to-point communication for large message counts, *due to queue search costs*

Form Communication Pattern with Locality-Awareness

- Normally send all indices to each process from which you want to receive data
- Instead, send a single message to each node with all indices to be send to each process (plus sizes of each message)
- Andrew: no poster, but paper on Arxiv
  - A Locality-Aware Sparse Dynamic Data Exchange (https://arxiv.org/abs/2308.13869v1)
- In MPI Advance, but needs an MPI interface



Form Communication Package : Suitesparse Matrices



tumorAntiAngiogenesis_4.mtx

nd12k.mtx

msc01050.mtx

Irregular Communication Steps

Point-to-Point Communication :

1. Form communication package
2. MPI_Send_init(s)
3. MPI_Recv_init(s)
4. Iterative MPI_Startall/MPI_Waitall

Neighbor Collective :

1. Form communication package
2. MPI_Dist_graph_create_adjacent
3. MPI_Neighbor_alltoallv_init
4. Iterative MPI_Start/MPI_Wait

Topology Communicator

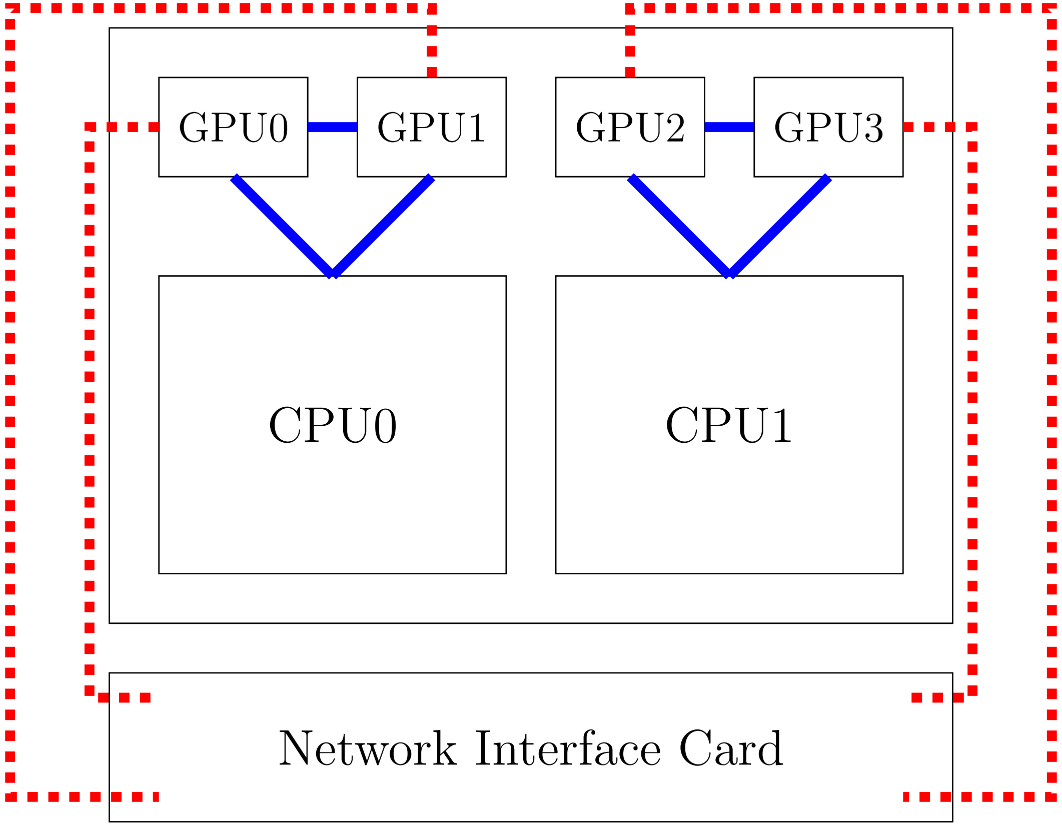
- MPI_Dist_graph_create_adjacent : creates a new communicator with a topology attached
  - Already know topology, pass it to this method
  - **All this method needs to do : take communication pattern information and store it**
  - Depending on implementation, currently **very expensive**

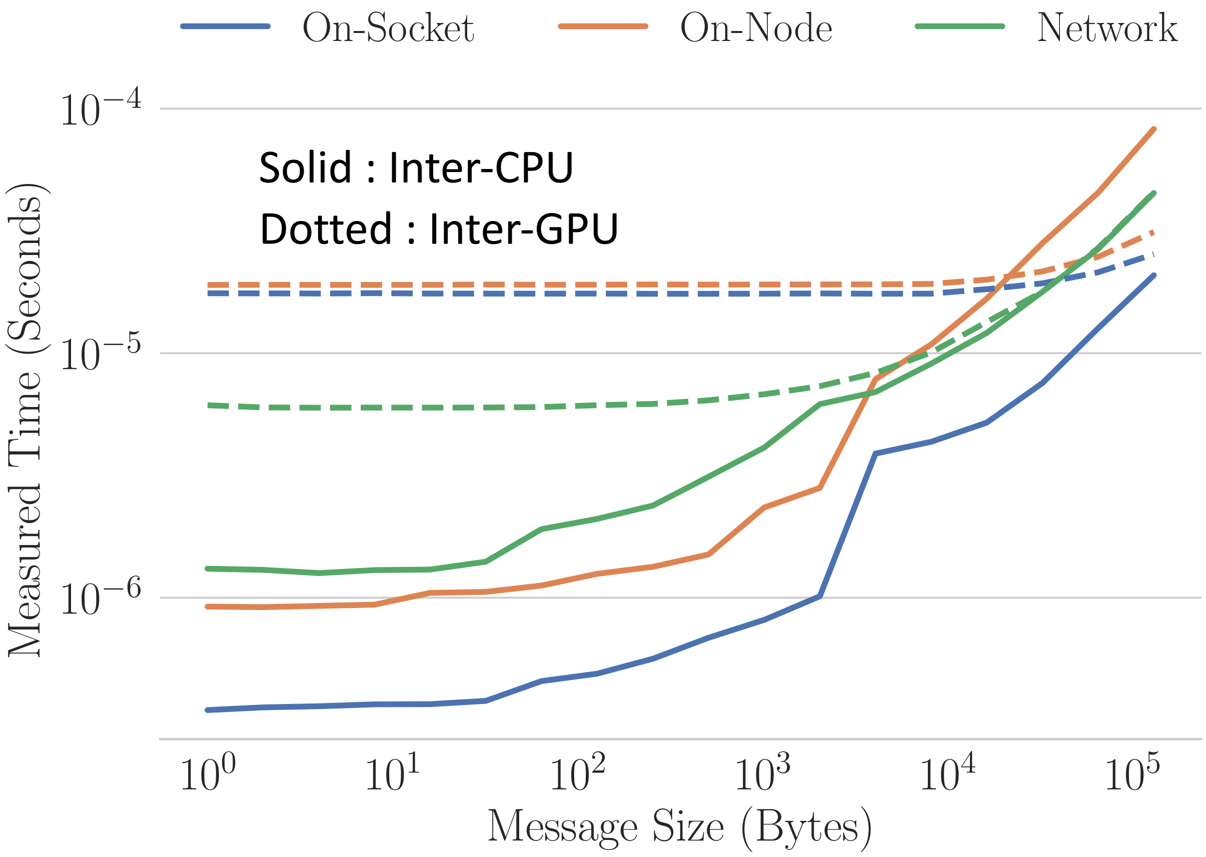
Hackathon : Topology in MPI Advance

- MPI Advance doesn’t need to follow the MPI standard
- Topology object within MPI Advance, storing information without creating a new communicator
- Reduces overhead of neighbor collectives

- *Related issue : Neighborhood collectives only go one direction*

Heterogeneous Architectures : Lassen





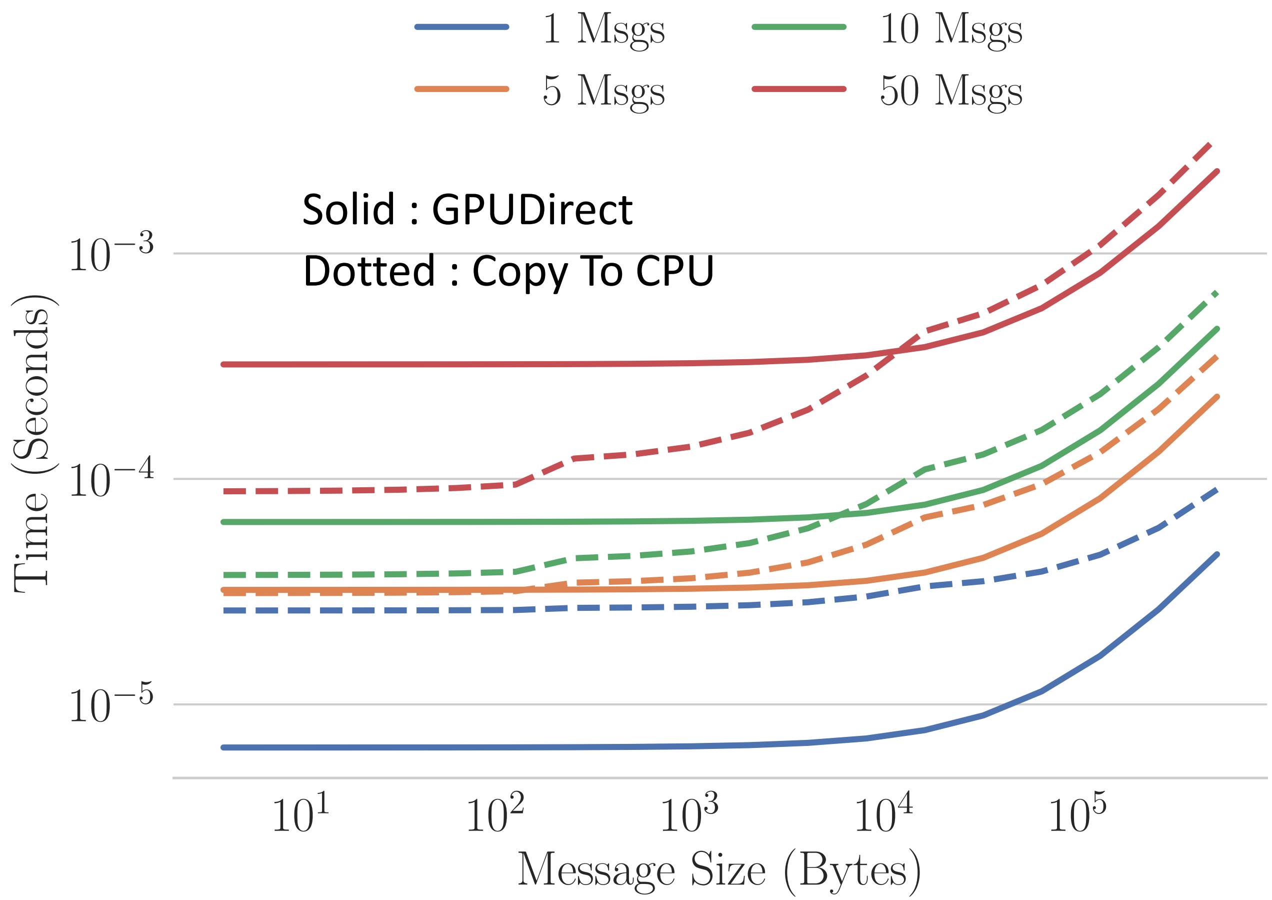
Solid : Inter-CPU

Dotted : Inter-GPU

Communication on Heterogeneous Architectures

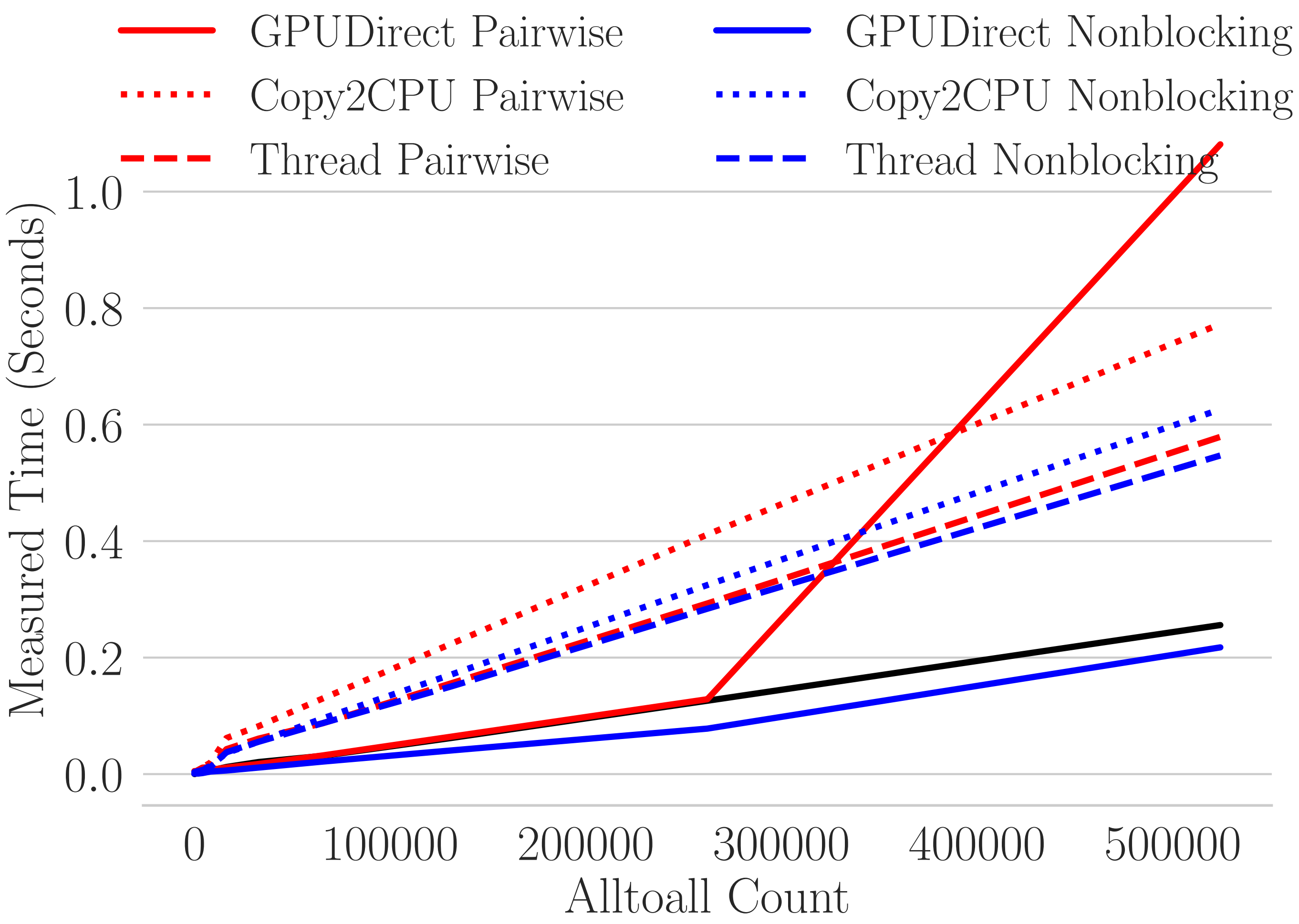
Multiple Messages

- Copy to CPU method :
  - Copy all data from GPU to CPU
  - Send many messages between CPUs
  - Copy all data to destination GPU
- Further optimization : use all available CPU cores



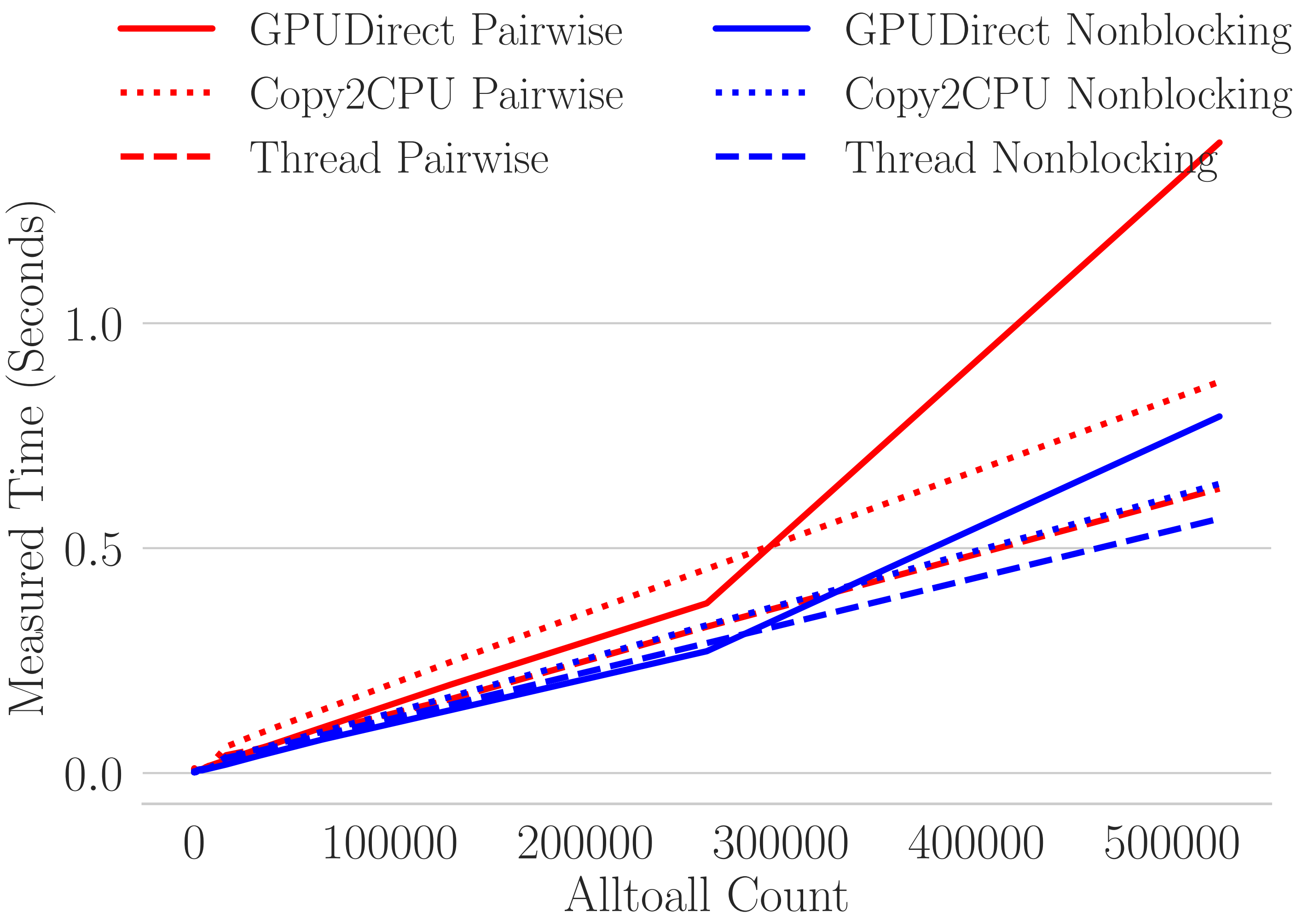
Key Takeaway : When sending large number of messages, cheaper to copy to CPU

MPI Advance Alltoall



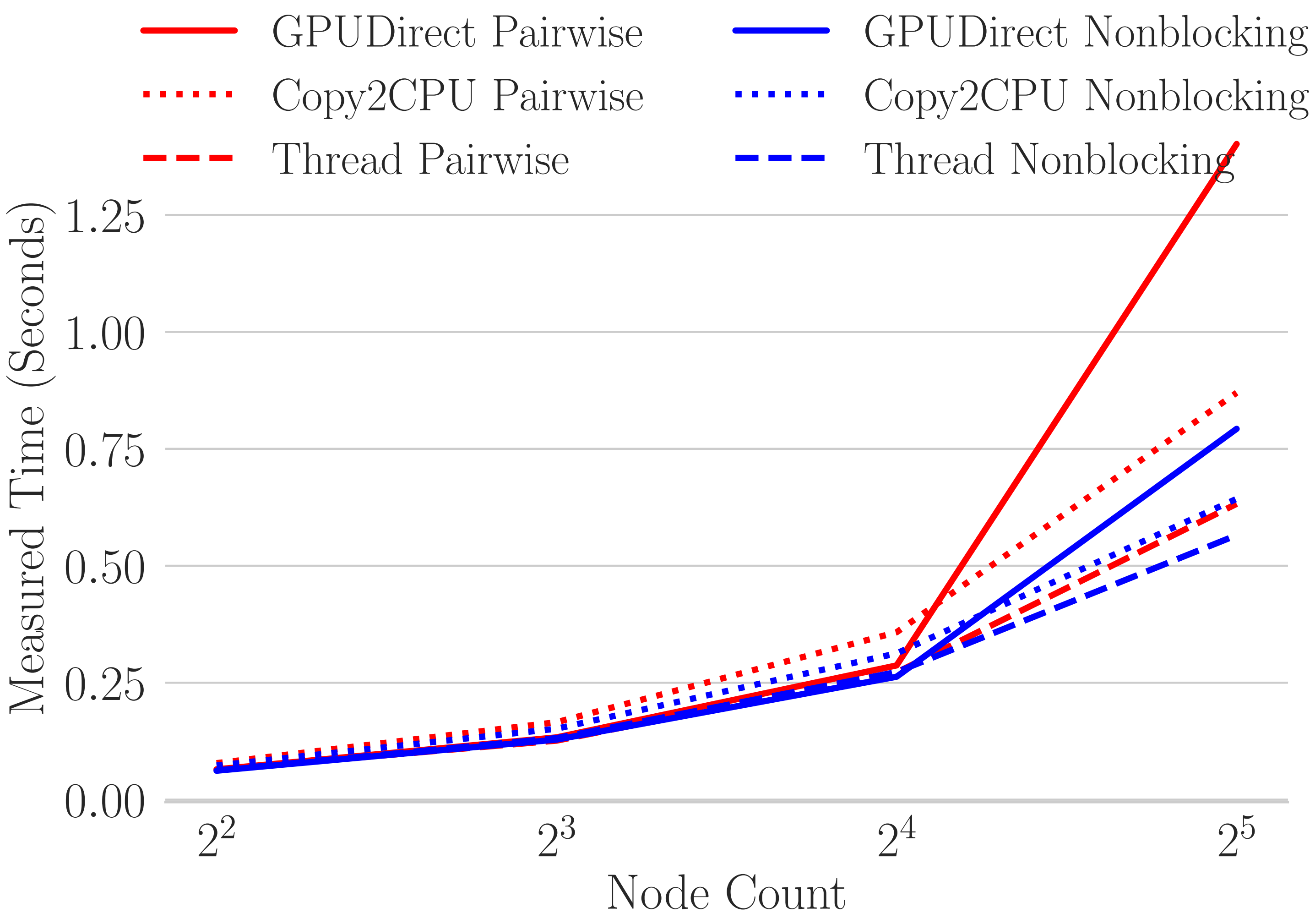
- 32 Nodes of Lassen
- Black line : SpectrumMPI
- Pairwise Exchange :
  - Send to / receive from one process at a time
- Nonblocking :
  - Isend/Irecv all messages at once
- Threaded : Copy GPU to CPU, launch threads, each send portion of messages

MPI Advance Alltoall



- 32 Nodes of Lassen
- Added cudaMallocHost to GPUDirect versions
- **Need to use persistent collectives**

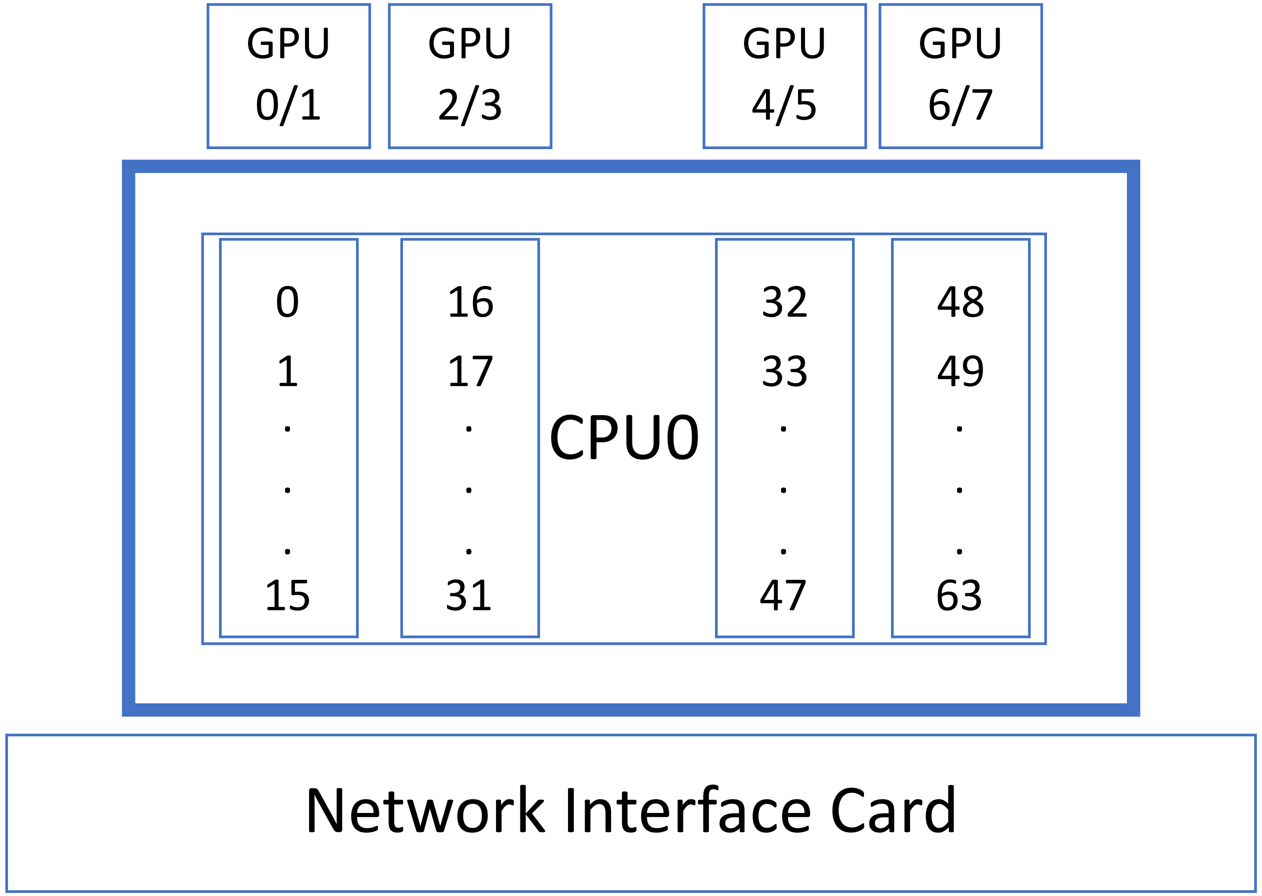
MPI Advance Alltoall



- Scaling study
- Added cudaMallocHost to GPUDirect versions
- **Need to use persistent collectives**

- Nicole’s Poster looks at benchmarking collective operations on Quartz
- Evelyn’s Poster looks at MPI_Alltoallv on Lassen

Heterogeneous Architectures : Tioga



GPU 0/1

GPU 2/3

GPU 4/5

GPU 6/7

0 1 . . . 15

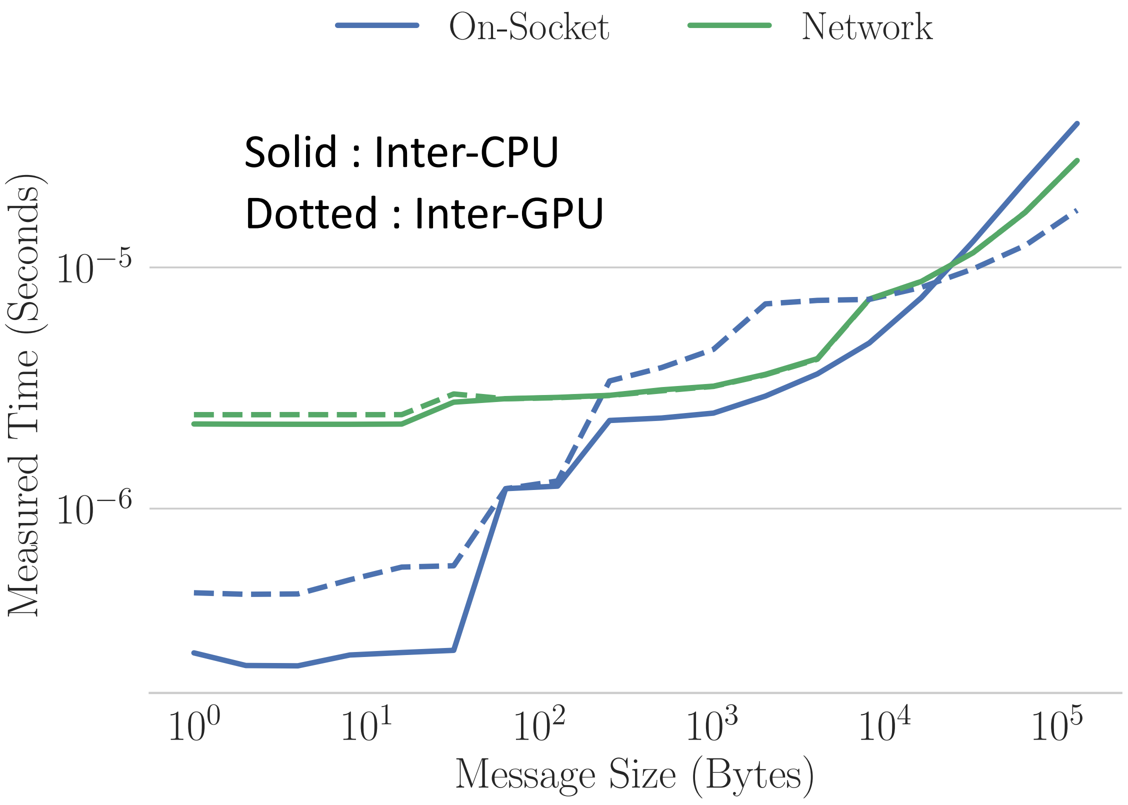
16 17 . . . 31

CPU0

32 33 . . . 47

48 49 . . . 63

Network Interface Card



Solid : Inter-CPU

Dotted : Inter-GPU

Other Current Work in Irregular Communication

- Students not funded by this project, but working under my supervision
  - Jackson Wesley is investigating methods to reduce queue search costs (and improve performance reproducibility)
  - Mike Adams is optimizing locality-aware neighbor collective algorithms on heterogeneous architectures
  - Louis Jencka is researching compression within sparse matrix-matrix multiplication (setup phase of AMG)

Other Irregular Communication Research

- Shannon Kinkead is analyzing network traffic during collective operations using SST
- Sandia employee, working in SST group
- Hopes to look into predictive modeling
- Plans to hold a tutorial to teach others (me) how to use SST

Questions?