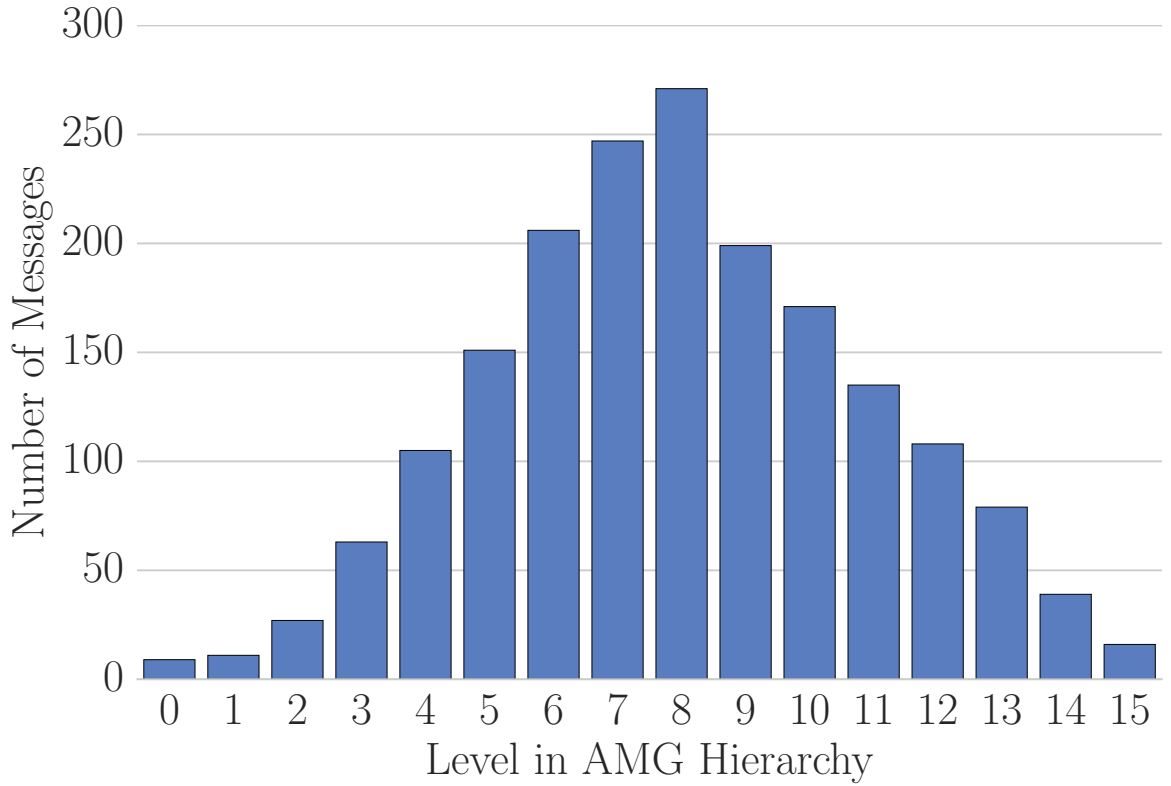
Irregular Boundary Exchanges

Prof. Amanda Bienz

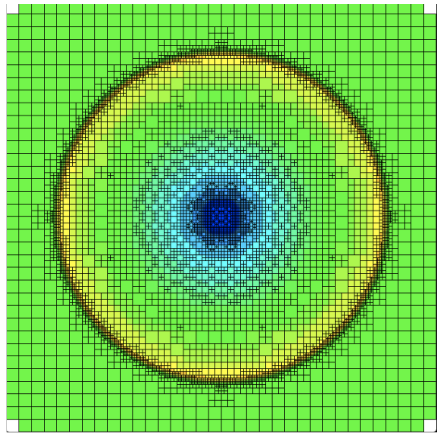
Motivation

- Applications with unstructured meshes, social network graphs, etc. often require boundary exchanges
  - Algebraic multigrid
  - Adaptive mesh refinement
- Large number of messages
- Must communicate with non-neighboring processes



Irregular Communication Codebases

Hypre
Algebraic Multigrid
LLNL



CLAMR
Adaptive Mesh Refinement
LANL

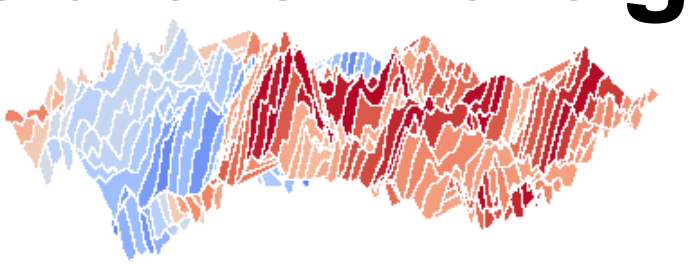
HIGRAD
Regular Meshes,
Implicit Solves
LANL

Other Irregular Communication
L7
Trilinos
xRage

Outline

- **Background**
- Year 1 Results
- Plans for Year 2
- Longer Term Future Directions

Algebraic Multigrid

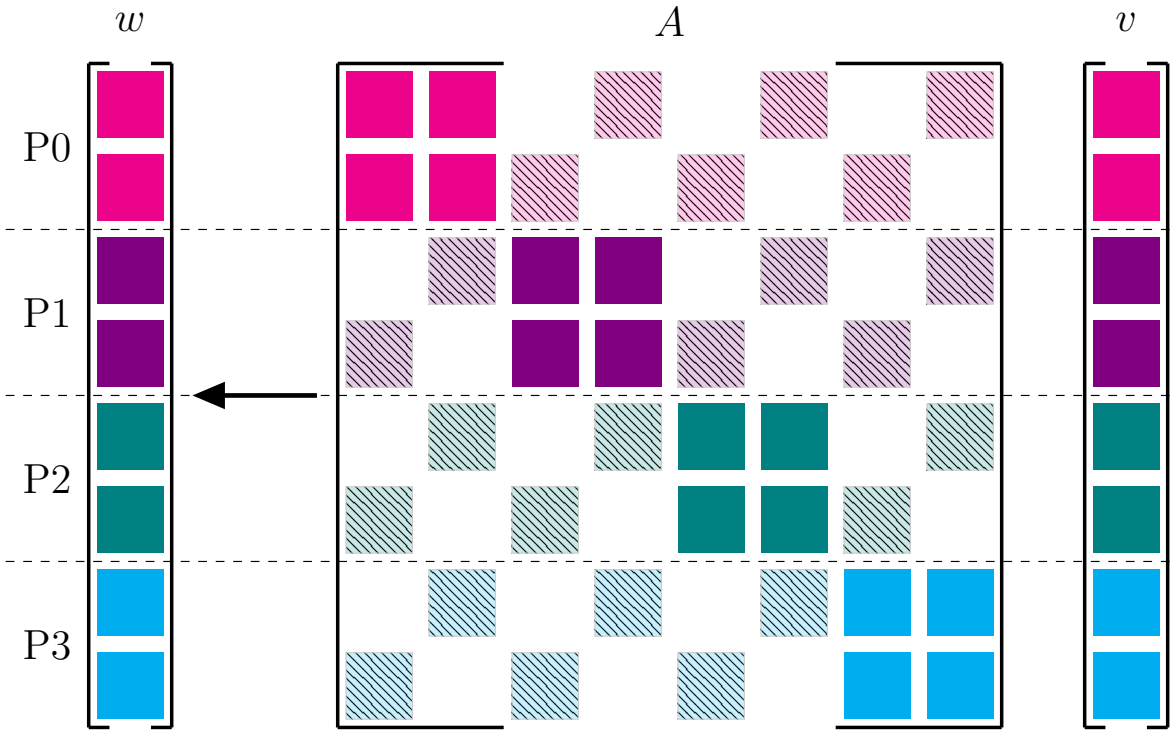


Relaxation
(Jacobi, Gauss-Seidel)

Restrict to coarser grid

- Two main operations on each level:
  - Sparse matrix-matrix multiply (SpGEMM)
  - Sparse matrix-vector multiply (SpMV)
- **Coarse matrices increase in density**

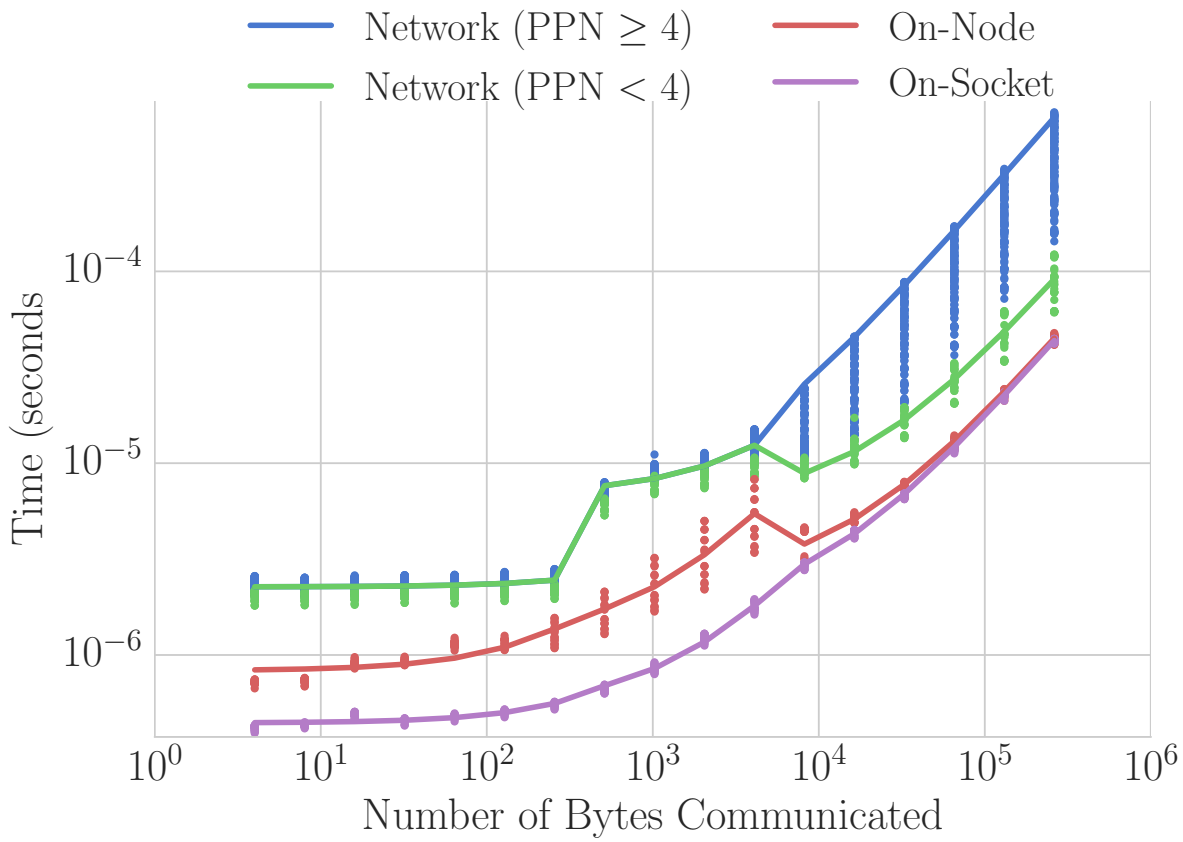
Parallel Sparse Matrix Operations

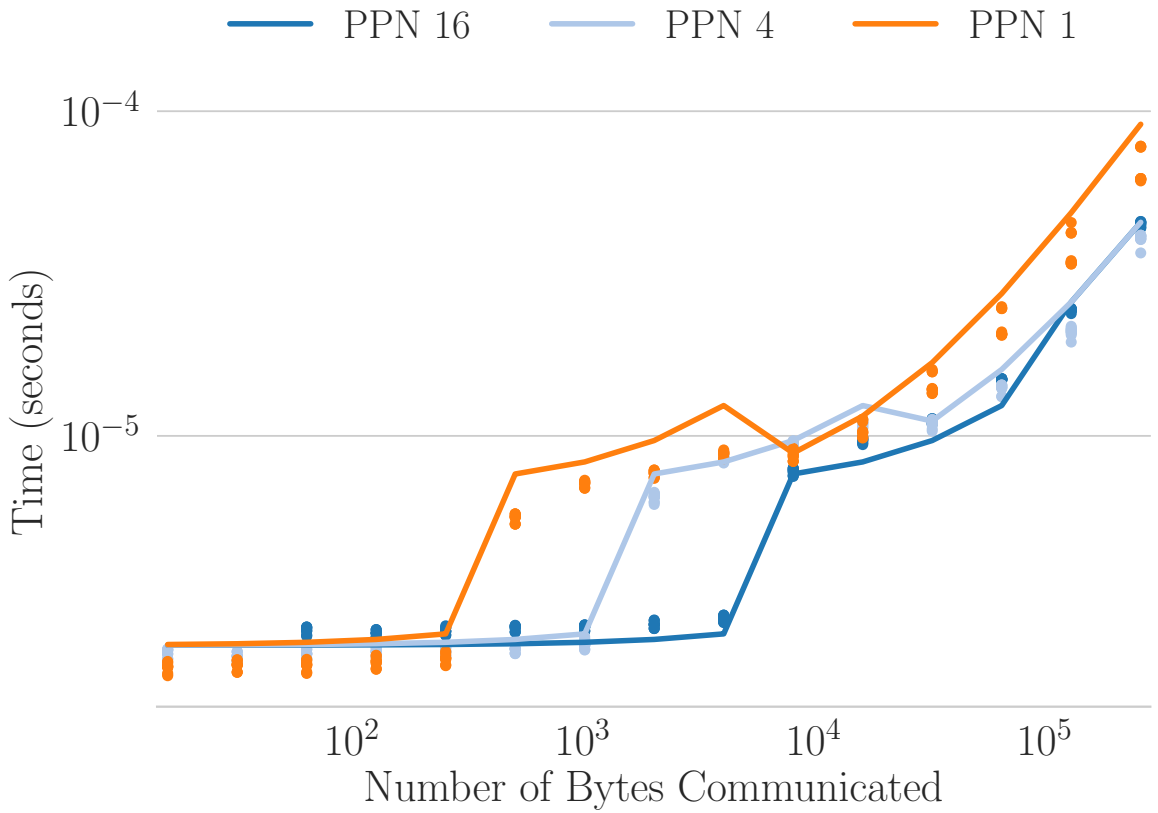


- Solid blocks : on-process
- Patterned blocks : off-process

Increased density → More patterned blocks → More communication

Data Movement Costs



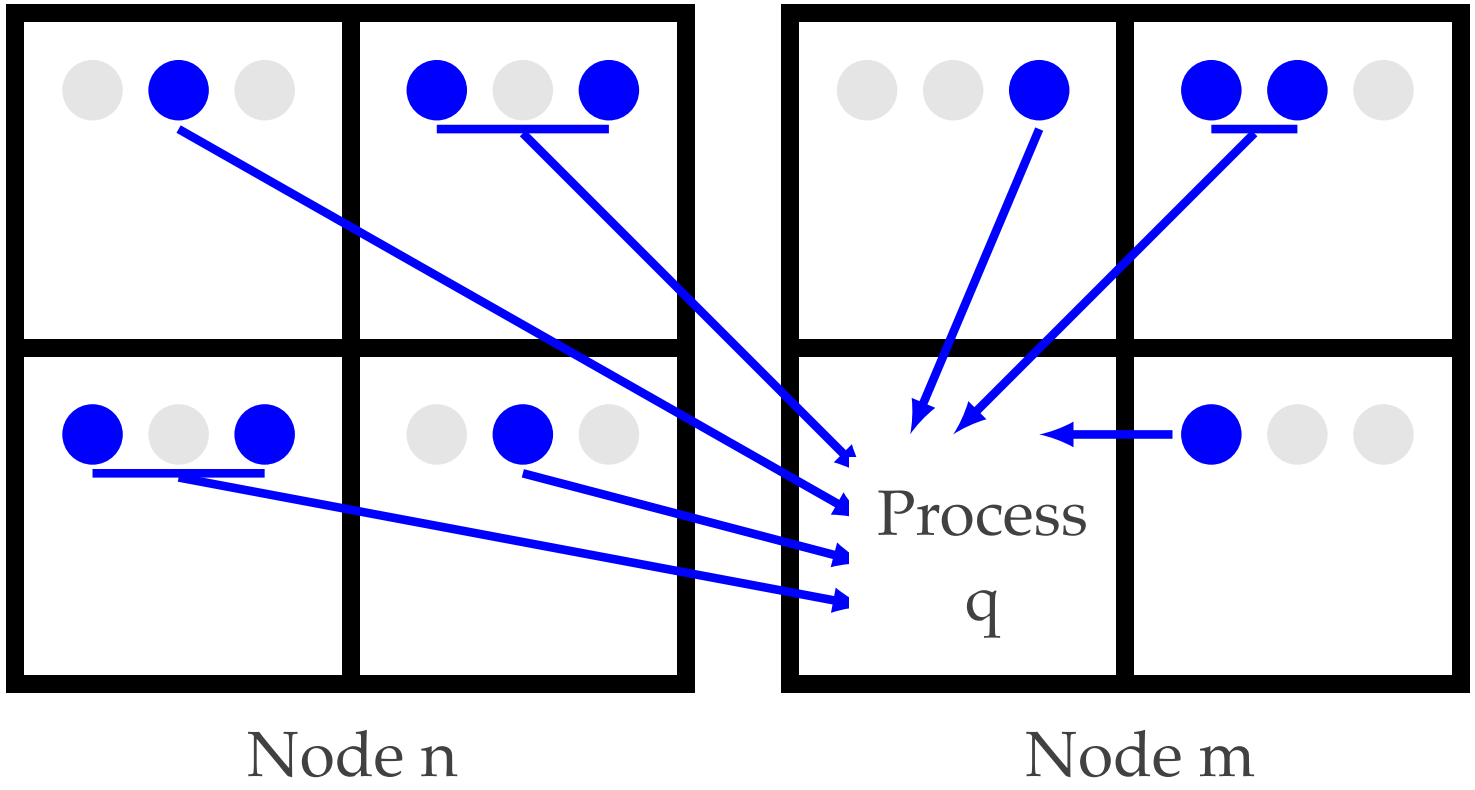


Locality-Aware Communication

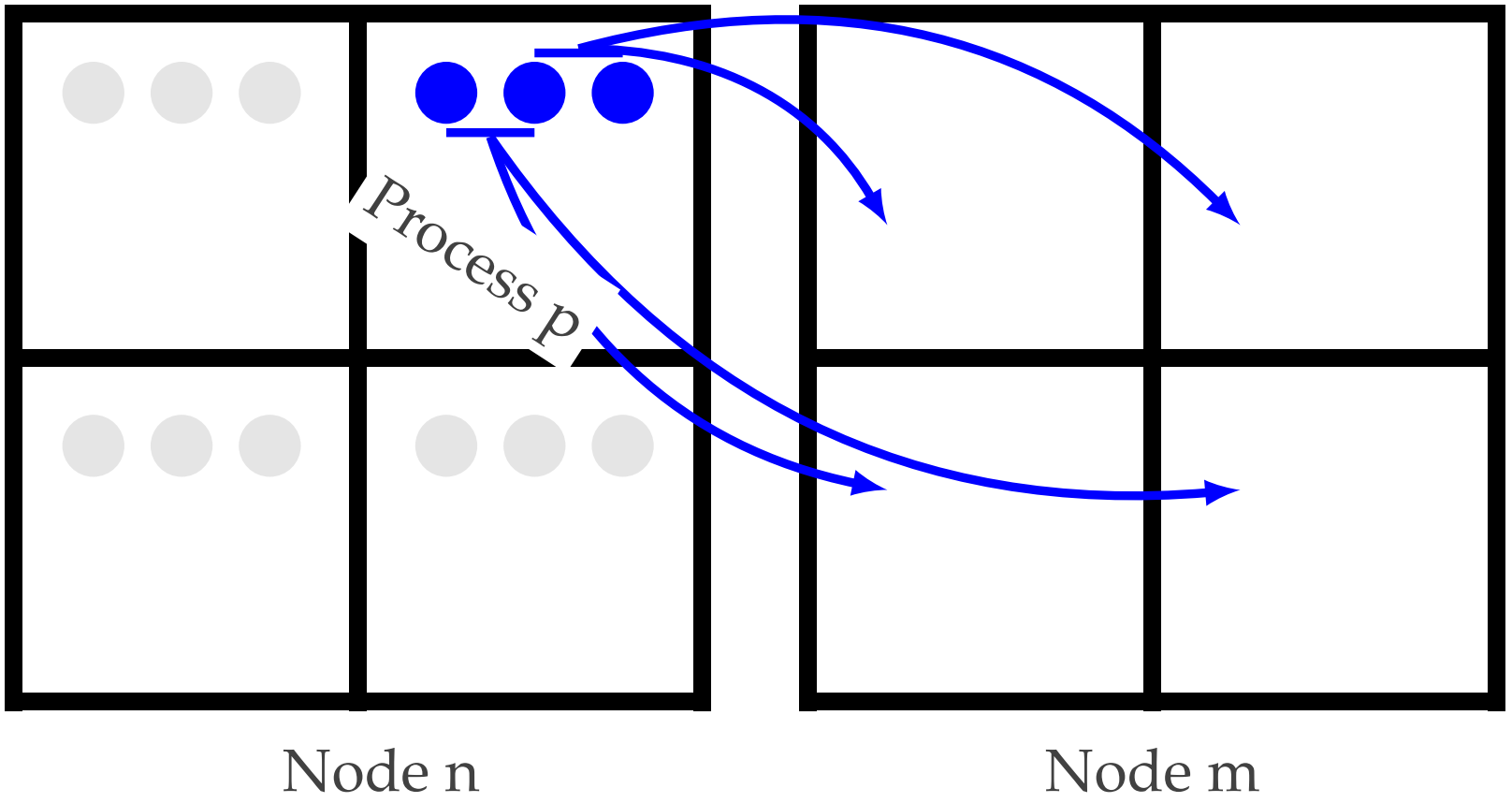
- Aggregate cheaper (e.g on-node) messages to reduce more expensive messages (e.g. off-node)

- Can reduce both the number and size of inter-node messages

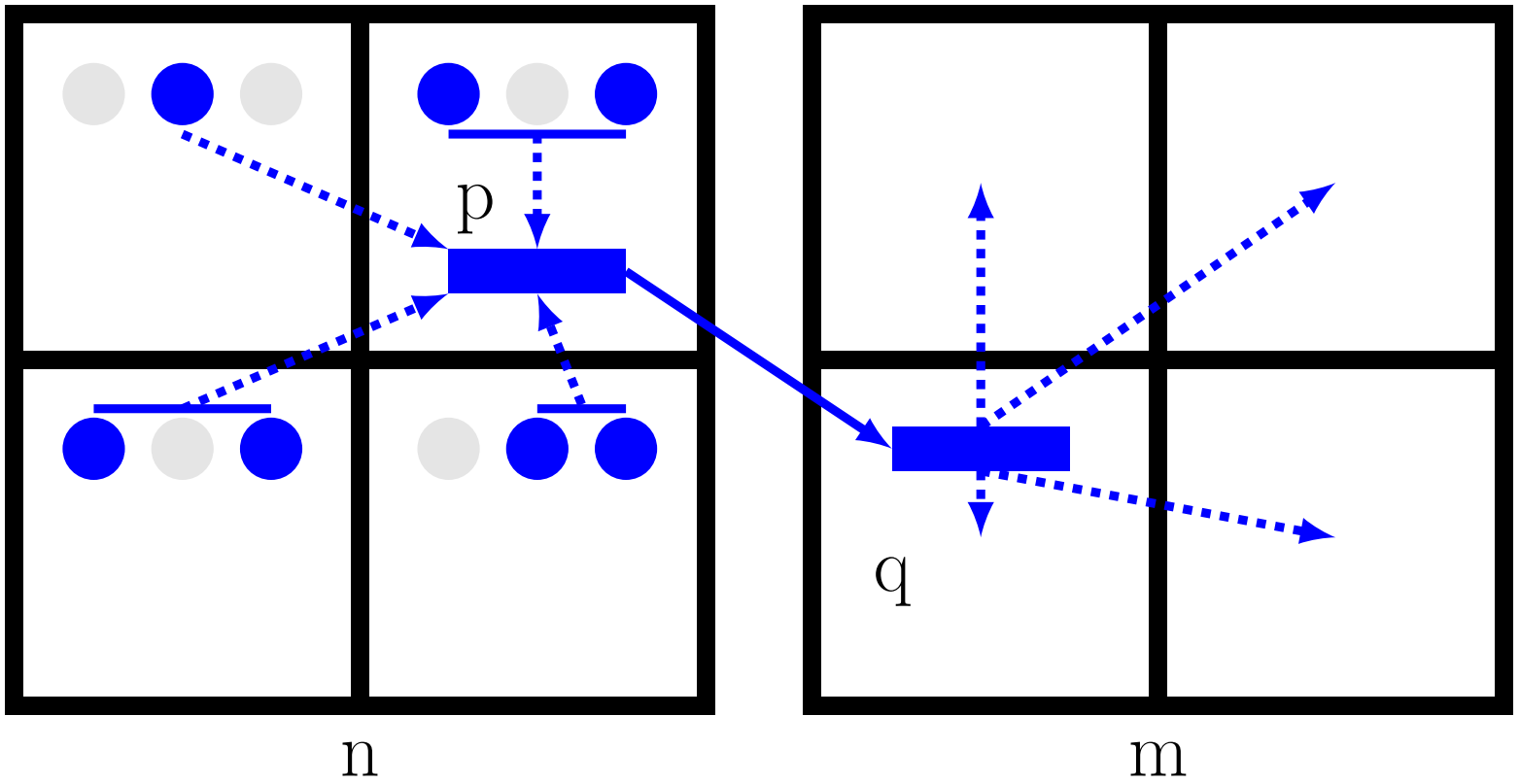
Standard Communication



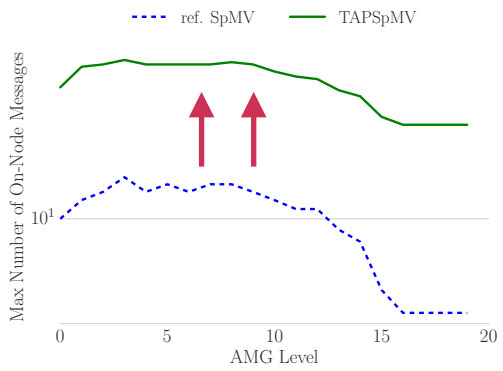
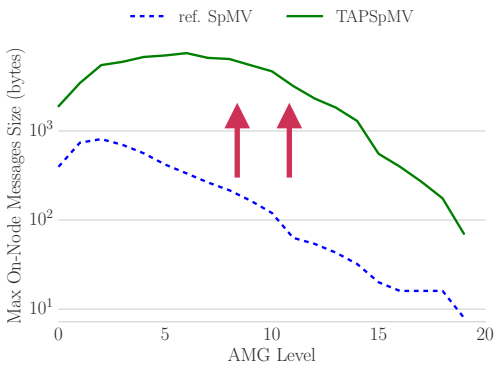
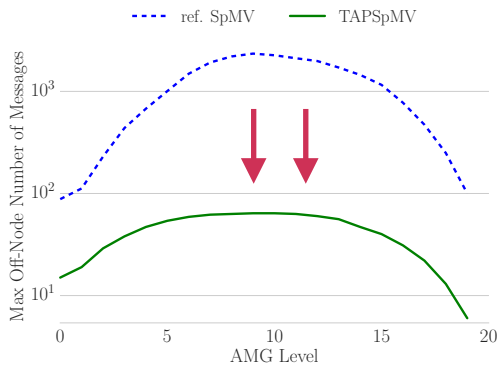
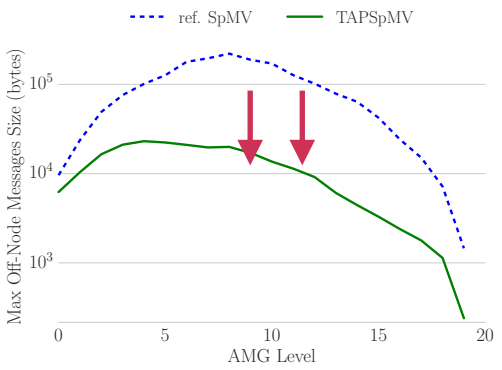
Standard Communication

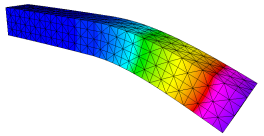


Locality-Aware Communication : Small Messages



Locality-Aware Communication

	Max Number of Messages	Max Size of Messages
Intra-node		
Inter-node		

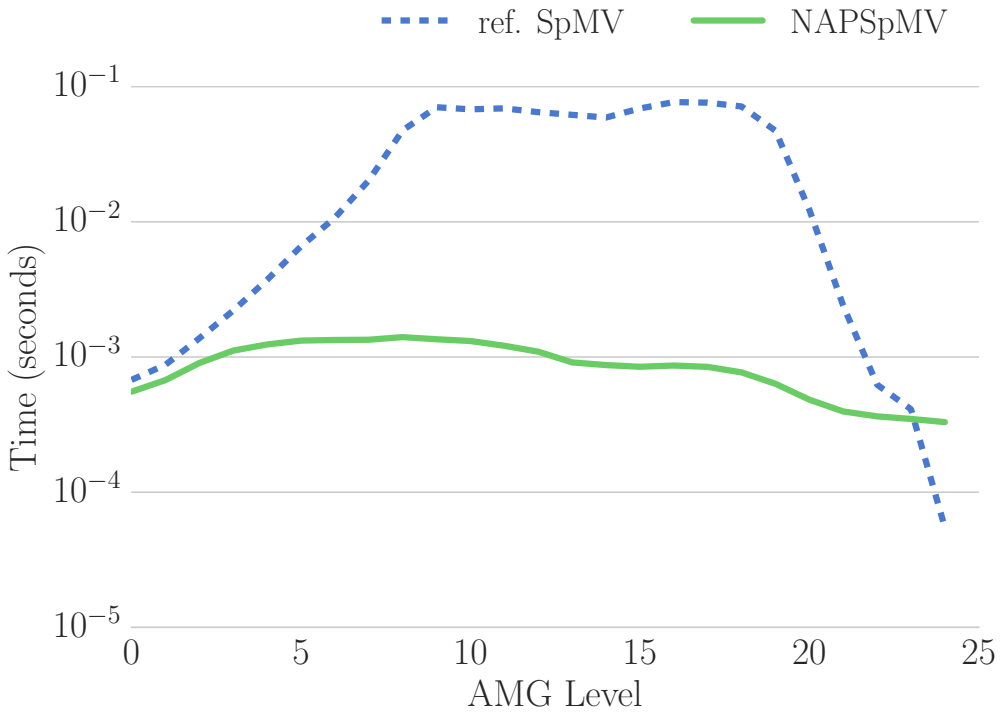


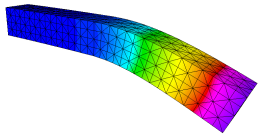
linear elasticity hierarchy
16,284 processes

Blue Dotted Lines :
Standard Communication

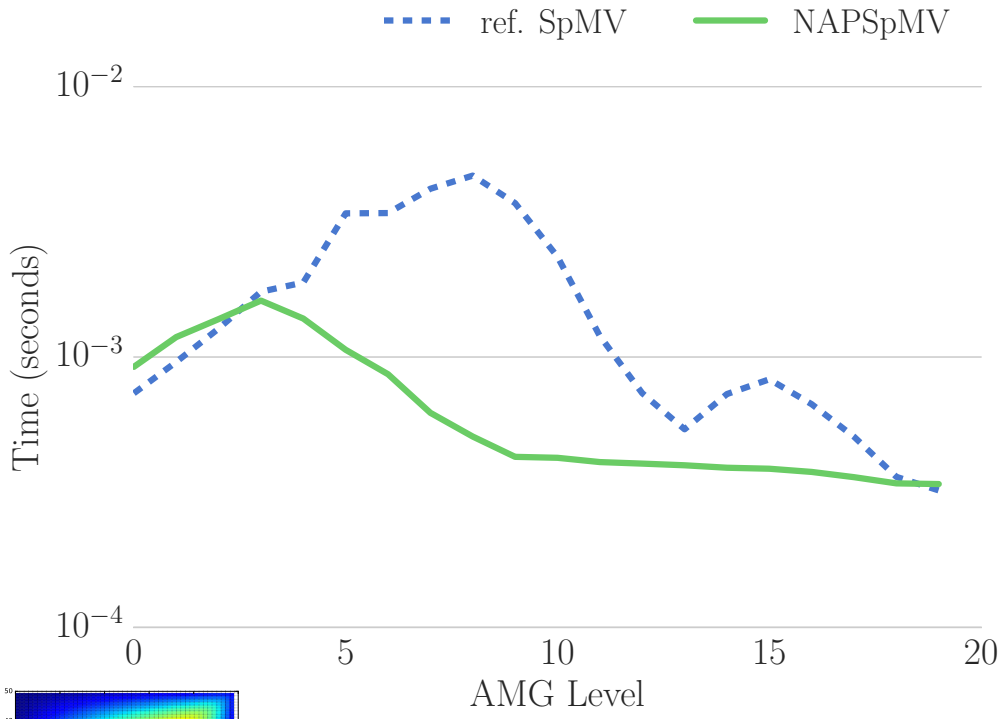
Green Lines:
Locality-Aware

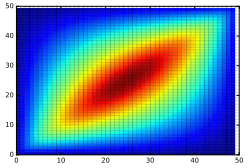
Locality-Aware SpMVs





Linear Elasticity (MFEM)

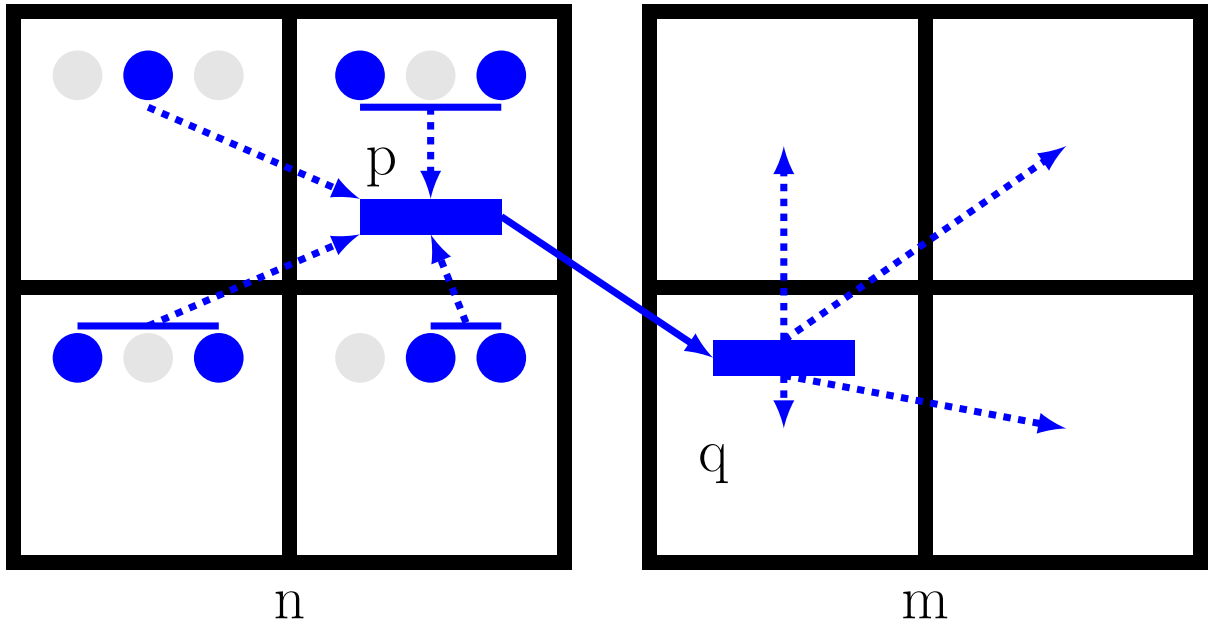


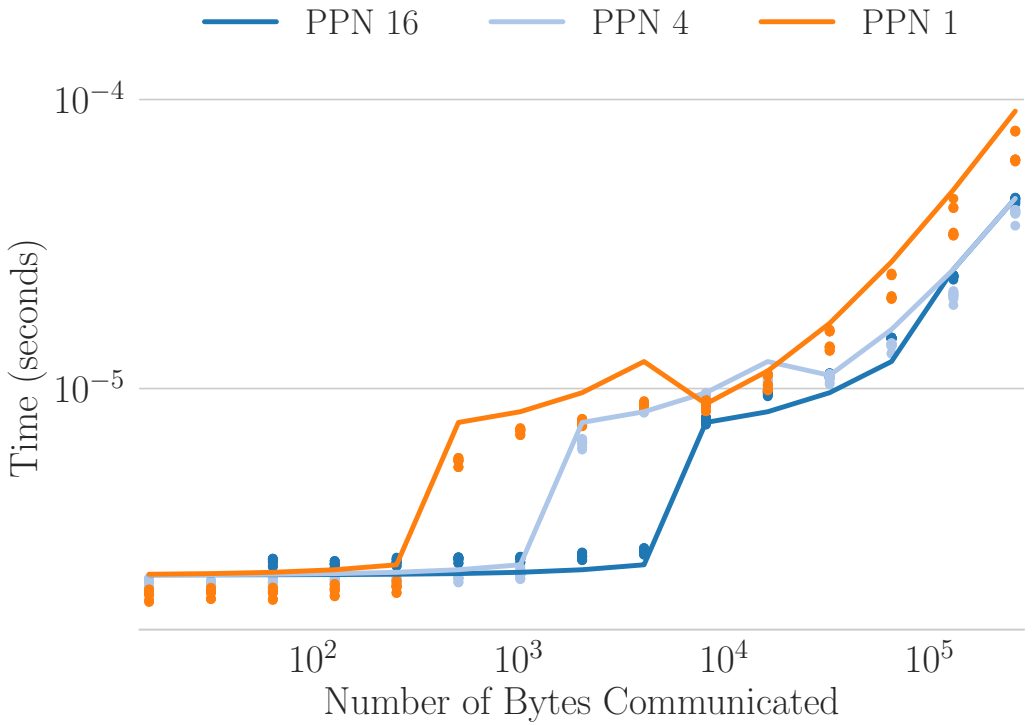


2D Rotated Anisotropic Diffusion

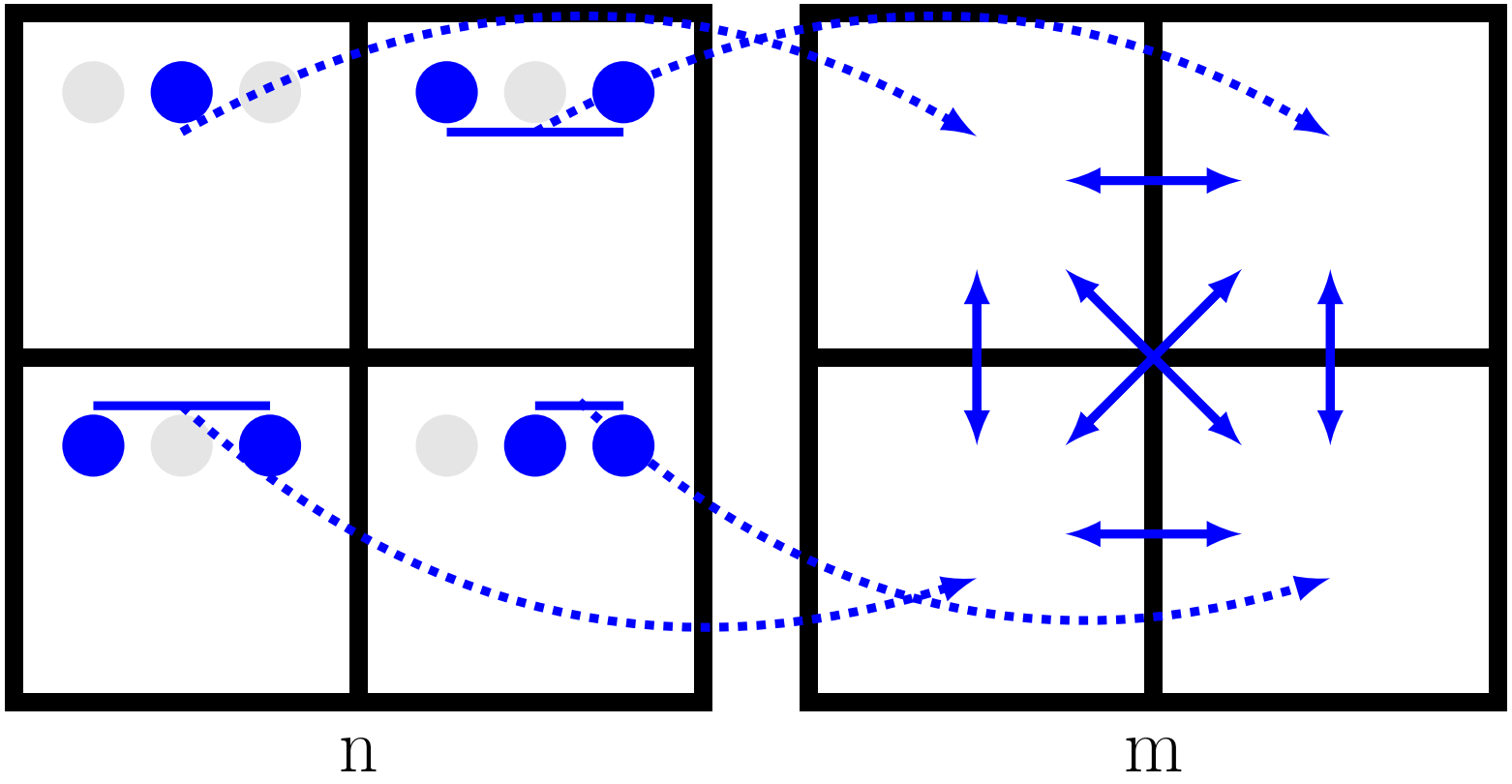
Node-Aware Sparse Matrix Vector Multiplication (https://doi.org/10.1016/j.jpdc.2019.03.016)

Locality-Aware Communication : Small Messages

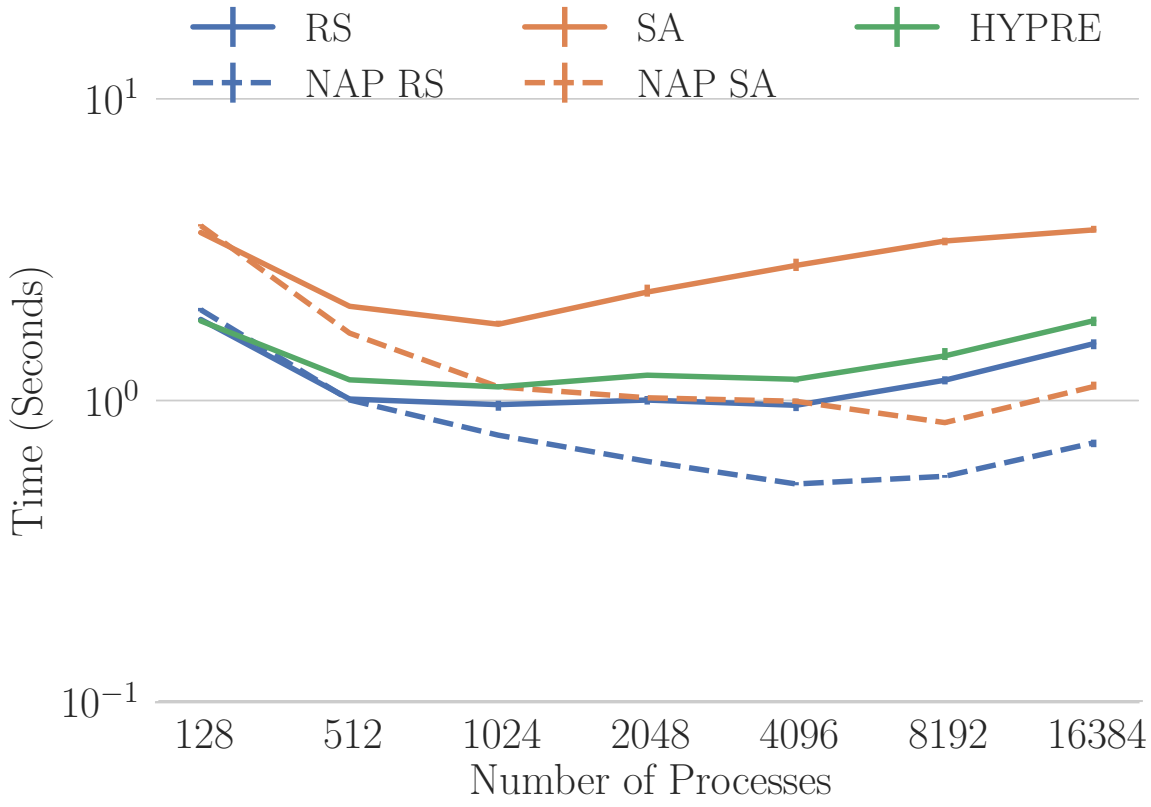




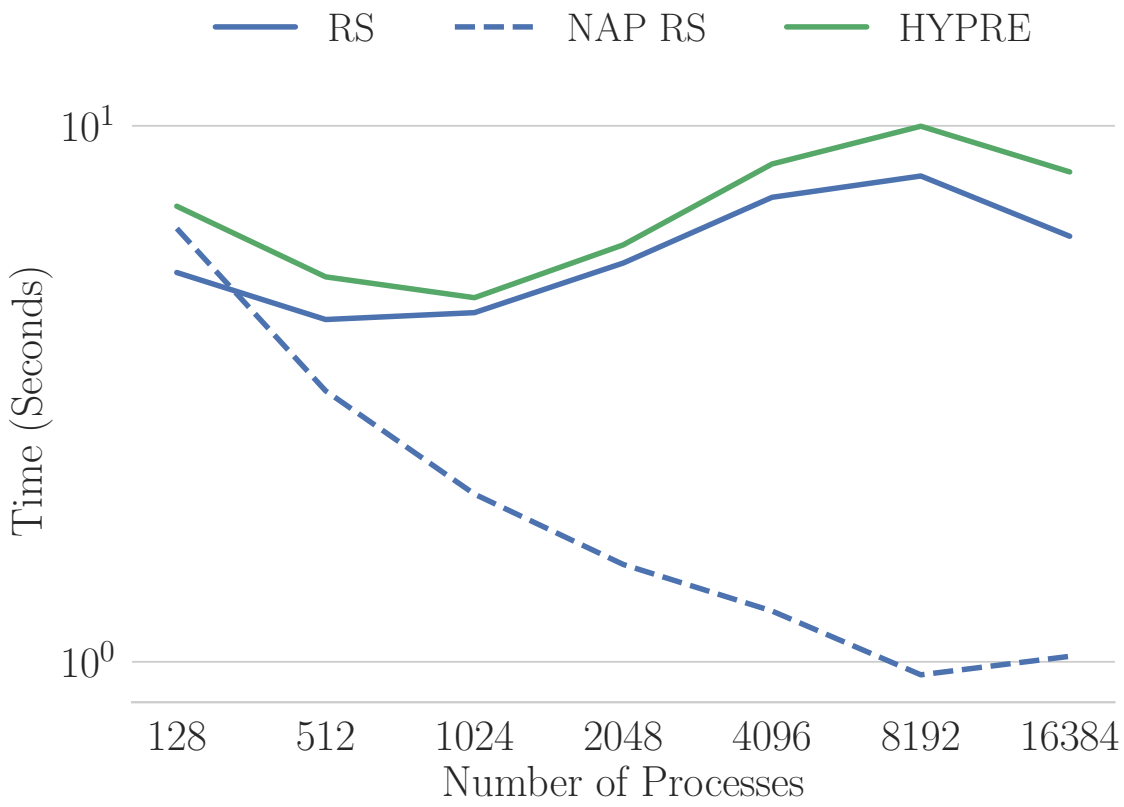
Locality-Aware Communication: Large Messages



Locality-Aware AMG

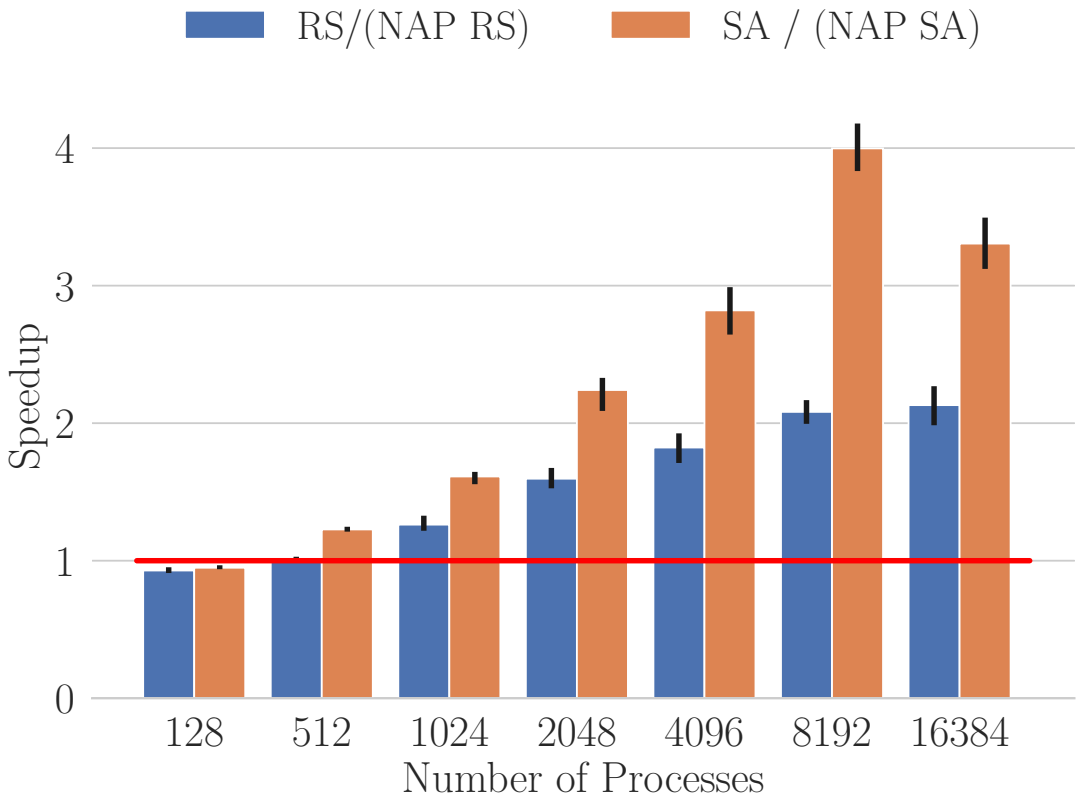


Grad-Div (MFEM)

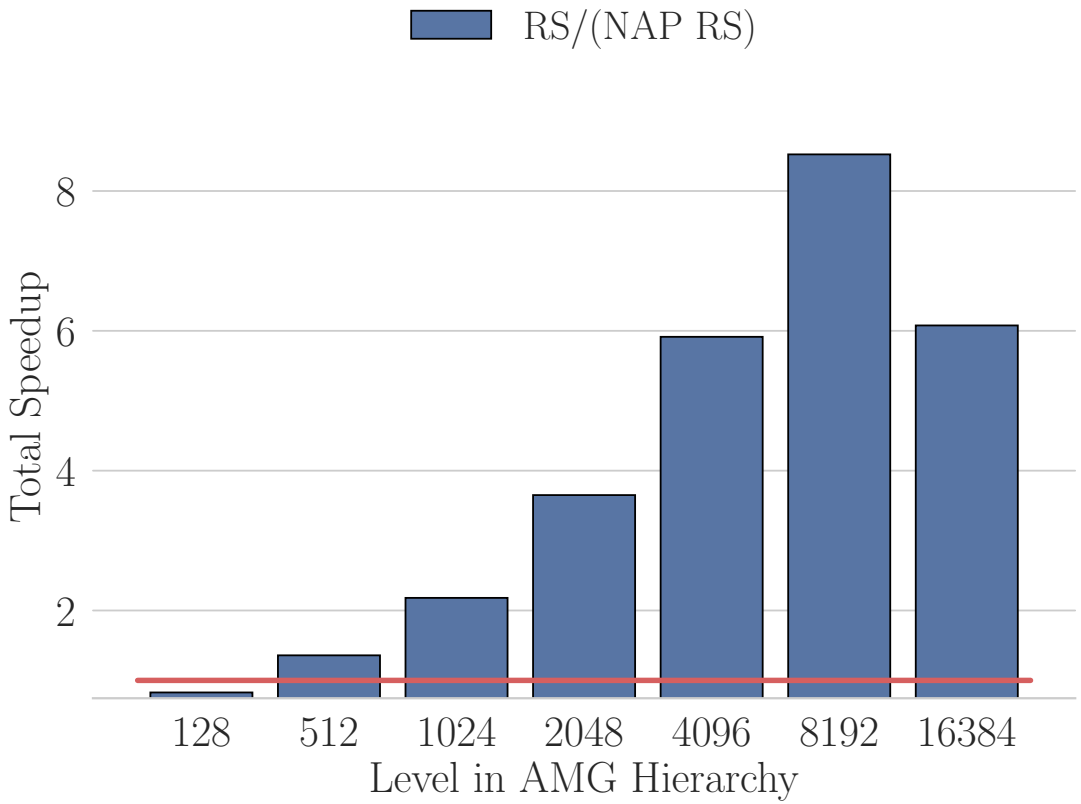


Laplacian (MFEM)

Locality-Aware AMG



Grad-Div (MFEM)



Laplacian (MFEM)

Reducing communication in algebraic multigrid with multi-step node aware communication (https://doi.org/10.1177/1094342020925535)

Outline

- Background
- **Year 1 Results**
- Plans for Year 2
- Longer Term Future Directions

MPI Advance

- **Goal : add aggregation into MPI library so optimizations can be applied to existing codebases (such as HYPRE)**

- More on MPI Advance : Derek Schafer's talk

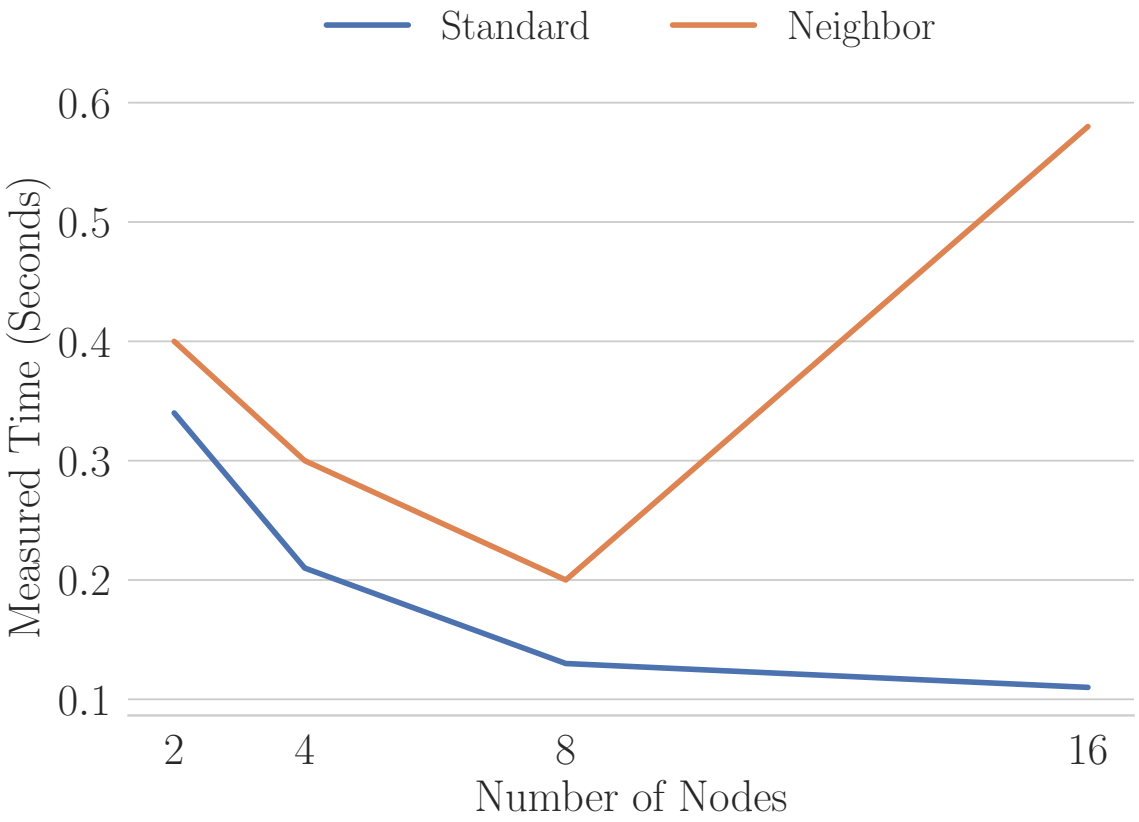
Neighborhood Collectives

```
int MPI_Dist_graph_create_adjacent(MPI_Comm comm_old,
                                   int indegree,
                                   const int sources[],
                                   const int sourceweights[],
                                   int outdegree,
                                   const int destinations[],
                                   const int destweights[],
                                   MPI_Info info,
                                   int reorder,
                                   MPI_Comm * comm_dist_graph)
```

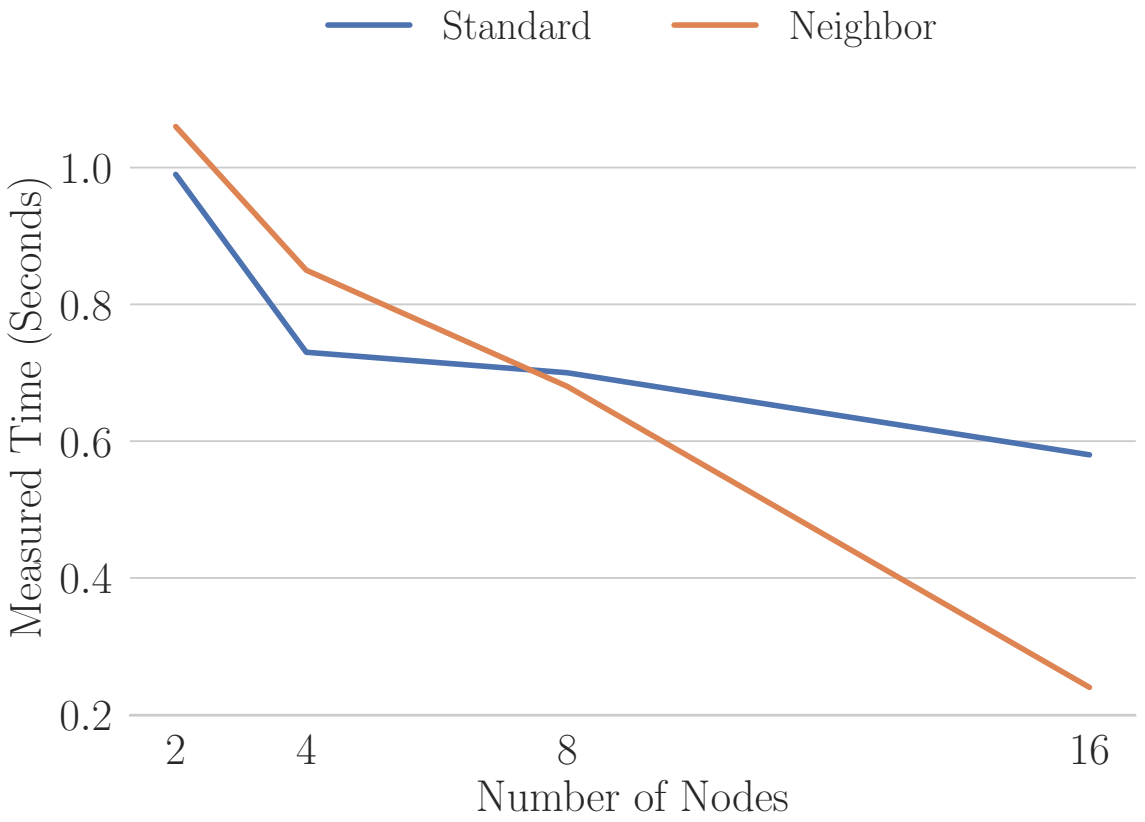
Neighborhood Collectives

```
int MPI_Neighbor_alltoallv(const void *sendbuf,
                           const int sendcounts[],
                           const int sdispls[],
                           MPI_Datatype sendtype,
                           void *recvbuf,
                           const int recvcounts[],
                           const int rdispls[],
                           MPI_Datatype recvtype,
                           MPI_Comm comm)
```

Neighborhood Collectives in HYPRE

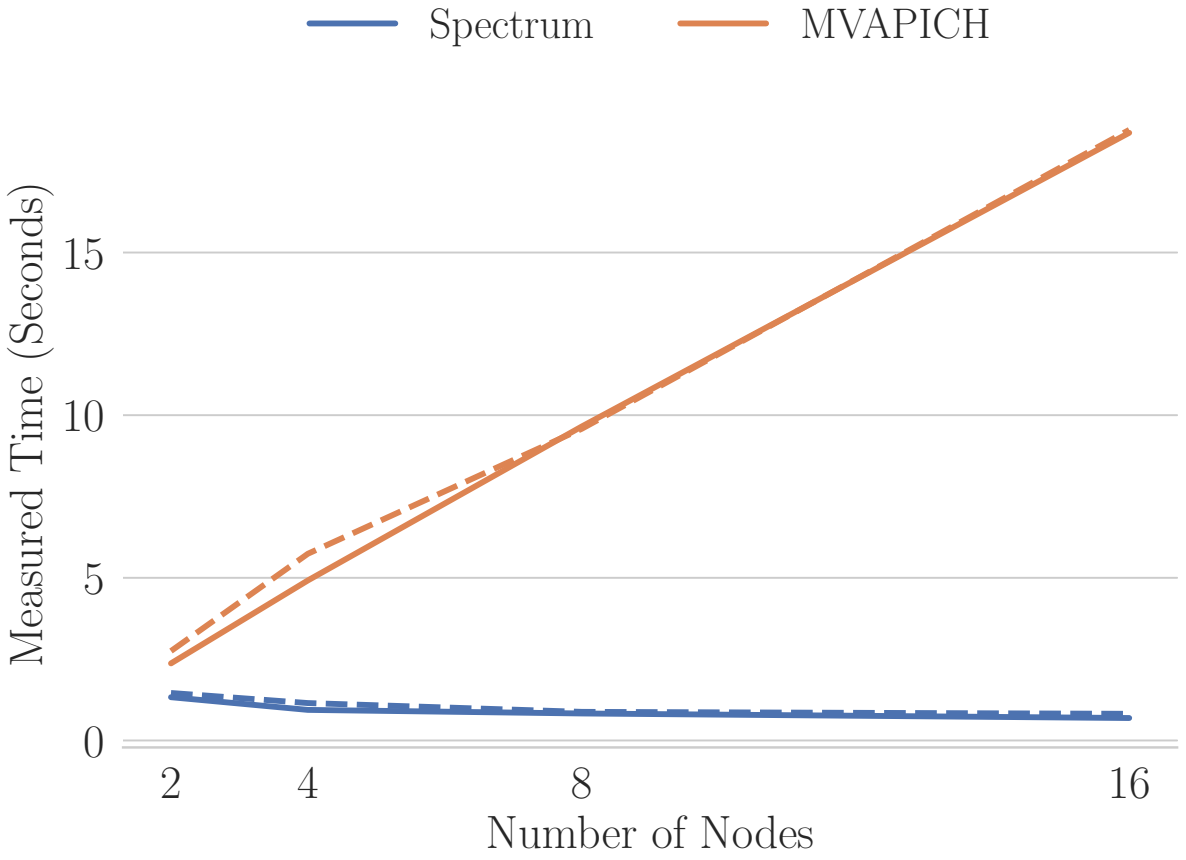


Setup Phase



Solve Phase

Lassen Spectrum MPI vs MVAPICH2-GDR



- Solid lines : standard HYPRE
- Dotted lines : HYPRE with neighborhood collectives

CLAMR

- More information on irregular communication with MVAPICH:
  - Tanner Broaddus's Talk

```
Profiling: Total CPU          time was   55.5333
-------------------------------------------------
Mesh Ops (Neigh+rezone+smooth+balance)   39.9882
Mesh Ops Percentage                      71.9815
```

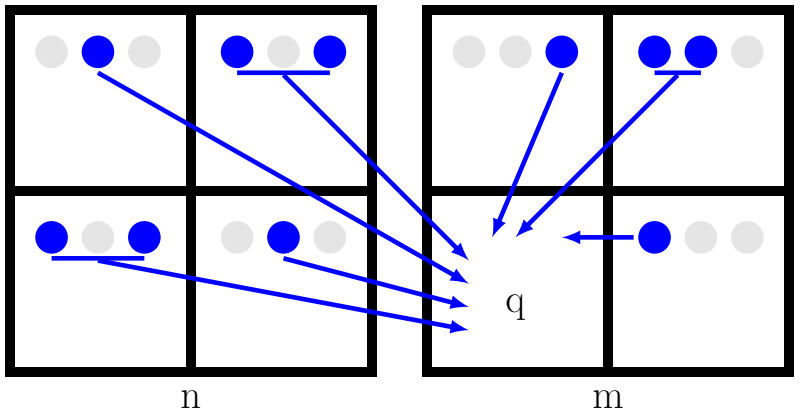
OpenMPI

```
Profiling: Total CPU          time was   62.8369
-------------------------------------------------
Mesh Ops (Neigh+rezone+smooth+balance)   47.2573
Mesh Ops Percentage                      75.1697
```

MVAPICH

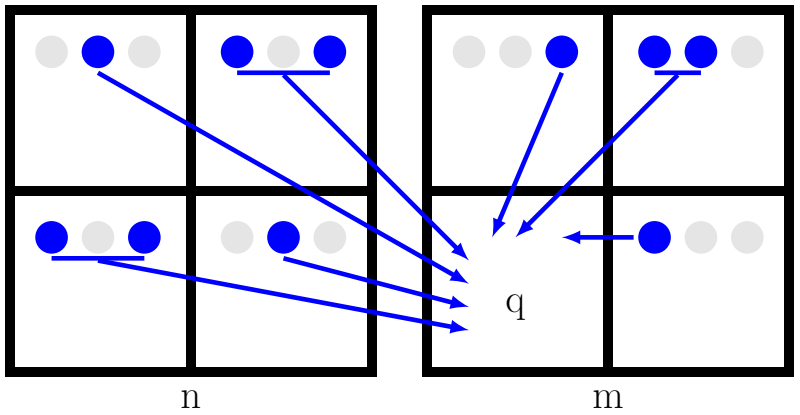
Aggregation in Neighborhood Collectives

```
int MPI_Dist_graph_create_adjacent(MPI_Comm comm_old,
                                   int indegree,
                                   const int sources[],
                                   const int sourceweights[],
                                   int outdegree,
                                   const int destinations[],
                                   const int destweights[],
                                   MPI_Info info,
                                   int reorder,
                                   MPI_Comm * comm_dist_graph)
```



Aggregation in Neighborhood Collectives

```
int MPIX_Dist_graph_create_adjacent(MPI_Comm comm_old,
                                    int indegree,
                                    const int sources[],
                                    const int sourceweights[],
                                    const int global_source_idx[],
                                    int outdegree,
                                    const int destinations[],
                                    const int destweights[],
                                    const int global_dest_idx[],
                                    MPI_Info info,
                                    int reorder,
                                    MPI_Comm * comm_dist_graph)
```

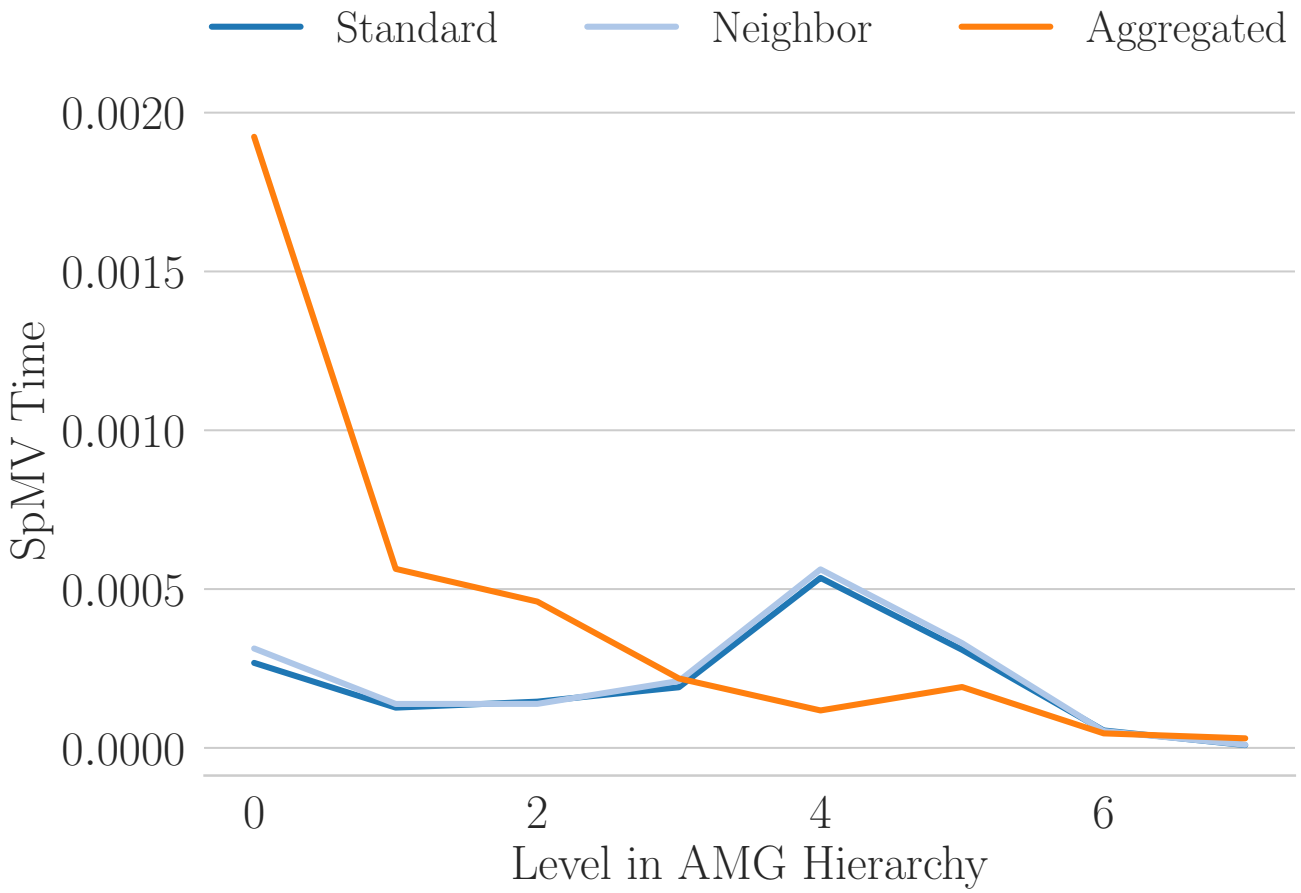


Aggregation in Neighborhood Collectives

```
int MPI_Neighbor_alltoallv(const void *sendbuf,
                           const int sendcounts[],
                           const int sdispls[],
                           MPI_Datatype sendtype,
                           void *recvbuf,
                           const int recvcounts[],
                           const int rdispls[],
                           MPI_Datatype recvtype,
                           MPI_Comm comm)
```

Note : if communicating sparse matrix (or any case where send counts not equivalent to weights in MPI_Dist_graph_create_adjacent), would like to change aggregation

Aggregated Neighborhood Collectives in Hypre



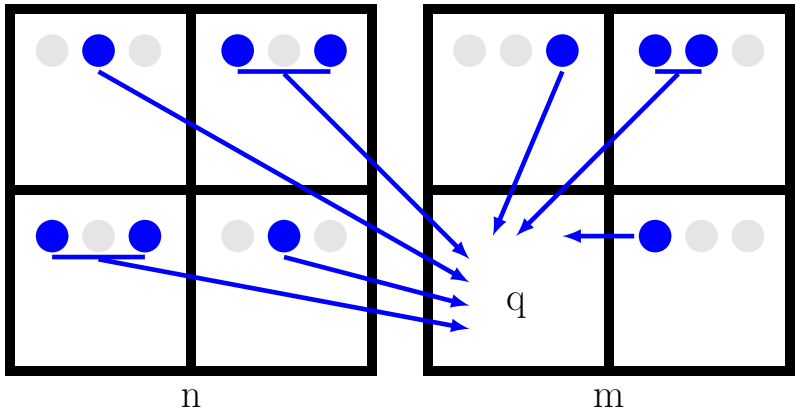
Outline

- Background
- Year 1 Results
- **Plans for Year 2**
- Longer Term Future Directions

Persistent Neighborhood Collectives

No setup for aggregation in MPI_Dist_graph_create_adjacent

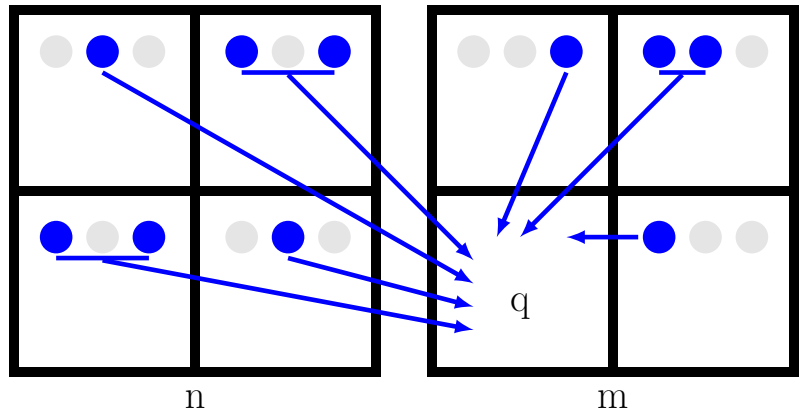
```
int MPI_Neighbor_alltoallv_init(const void *sendbuf,
                                const int sendcounts[],
                                const int sdispls[],
                                MPI_Datatype sendtype,
                                void *recvbuf,
                                const int recvcounts[],
                                const int rdispls[],
                                MPI_Datatype recvtype,
                                MPI_Comm comm,
                                MPI_Request* request)
```



Persistent Neighborhood Collectives

No setup for aggregation in MPI_Dist_graph_create_adjacent

```
int MPIX_Neighbor_alltoallv_init(const void *sendbuf,
                                 const int sendcounts[],
                                 const int sdispls[],
                                 const int global_send_idx[],
                                 MPI_Datatype sendtype,
                                 void *recvbuf,
                                 const int recvcounts[],
                                 const int rdispls[],
                                 const int global_recv_idx[],
                                 MPI_Datatype recvtype,
                                 MPI_Comm comm,
                                 MPI_Request* request)
```



Persistent Neighborhood Collectives

Aggregation setup in init method. Now, just communicate data every time MPI_Start is called.

```
int MPI_Start(MPI_Request* request)
int MPI_Wait(MPI_Request* request, MPI_Status* status)
```

Next Steps

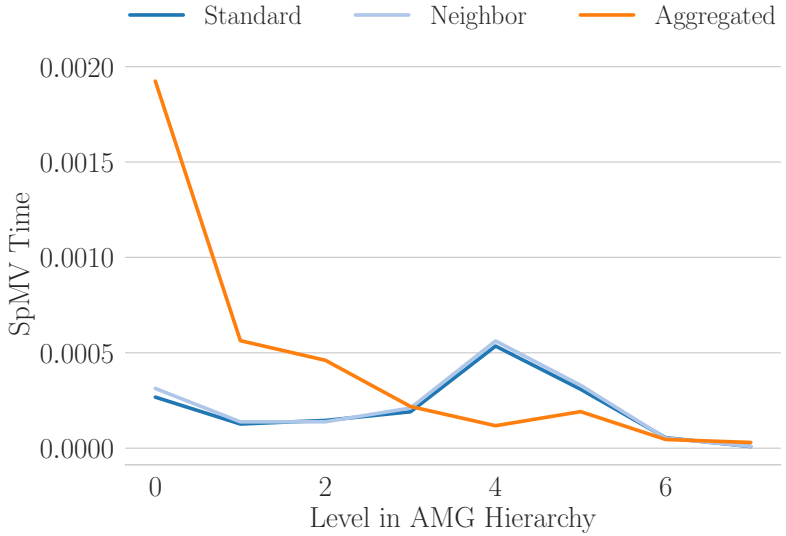
1. Analyze aggregated neighborhood collectives in Hypre
2. Analyze MPI_ vs MPIX_ API improvements
3. Analyze with different versions of MPI

Outline

- Background
- Year 1 Results
- Plans for Year 2
- **Longer Term Future Directions**

Longer Term Goals

1. Add two-step aggregation
2. Determine when different methods of communication should be used
3. Analyze overhead in dist_graph_create_adj
4. Analyze MPI advance improvements in HIGRAD implicit solves



Thanks for your time!

Questions?