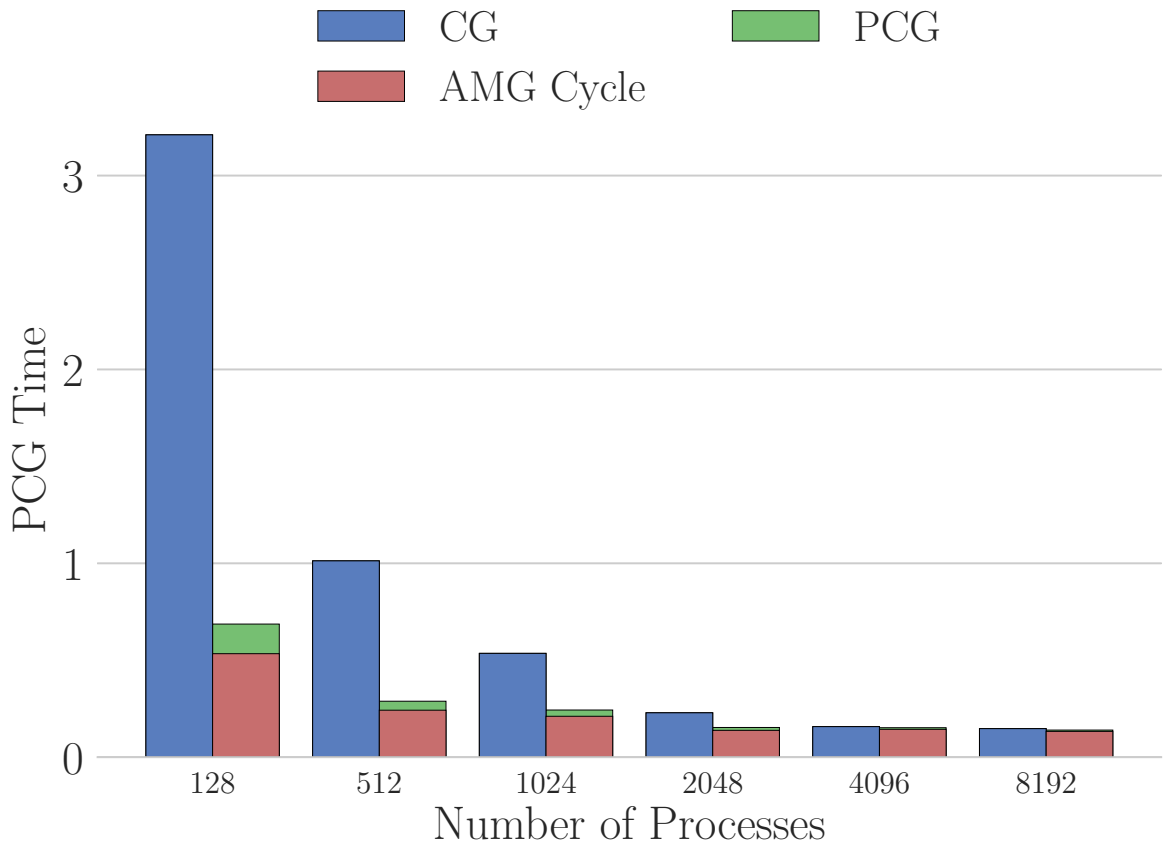
Improving Sparse Solvers with Locality-Aware Data Movement

Amanda Bienz

In collaboration with : Luke Olson, Bill Gropp

Motivation

- Algebraic multigrid (AMG) is an O(n) solver for sparse linear equations
- Typically, used as a preconditioner for Krylov methods (CG)
- Lacks parallel scalability



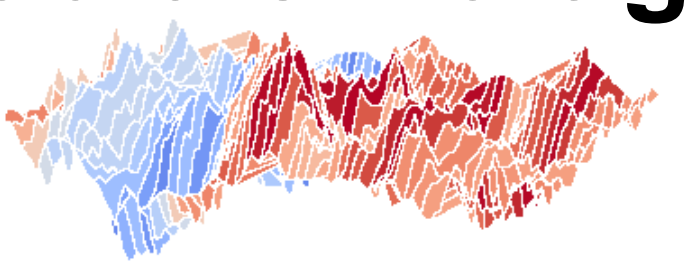
Algebraic Multigrid Codebases

Hypre
Lawrence Livermore

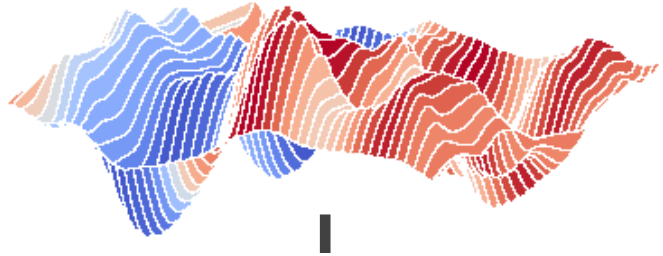
RAPtor

Muelu
Sandia

Algebraic Multigrid



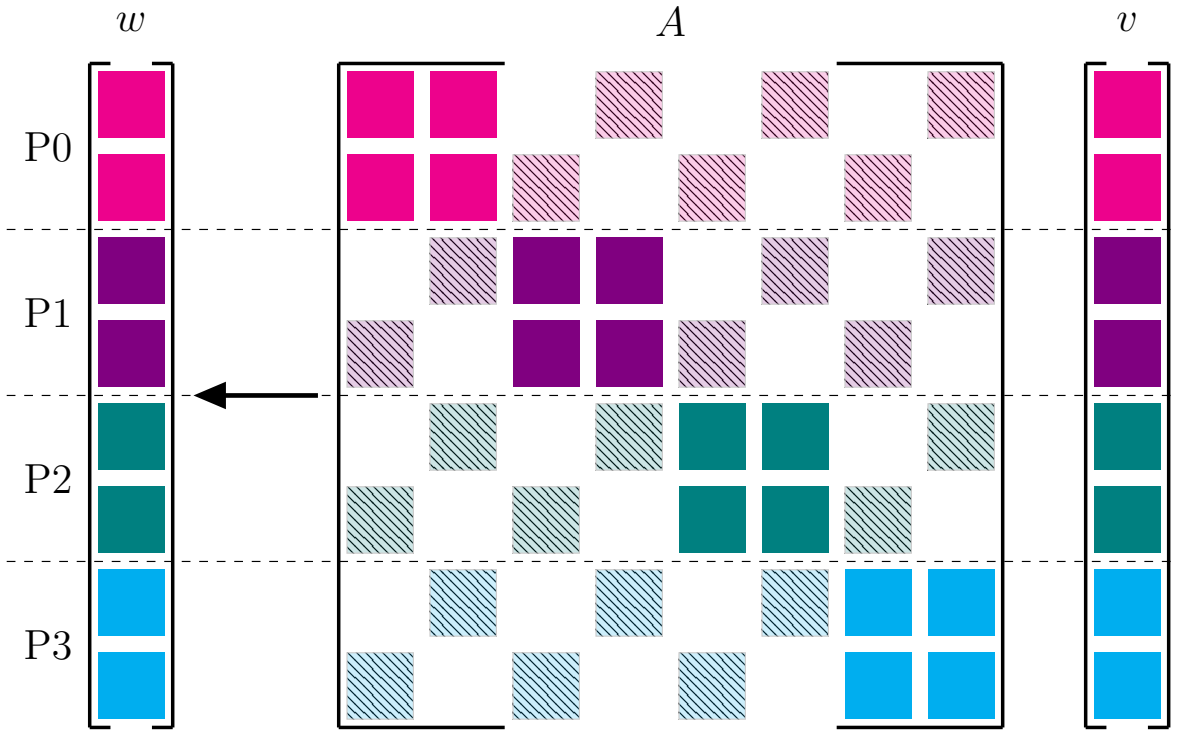
Relaxation
(Jacobi, Gauss-Seidel)



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**

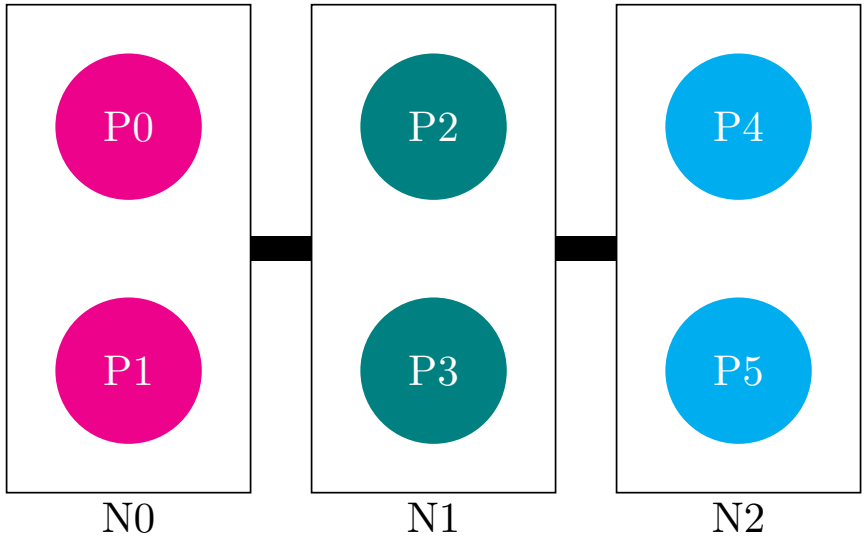
Parallel Sparse Matrix Operations



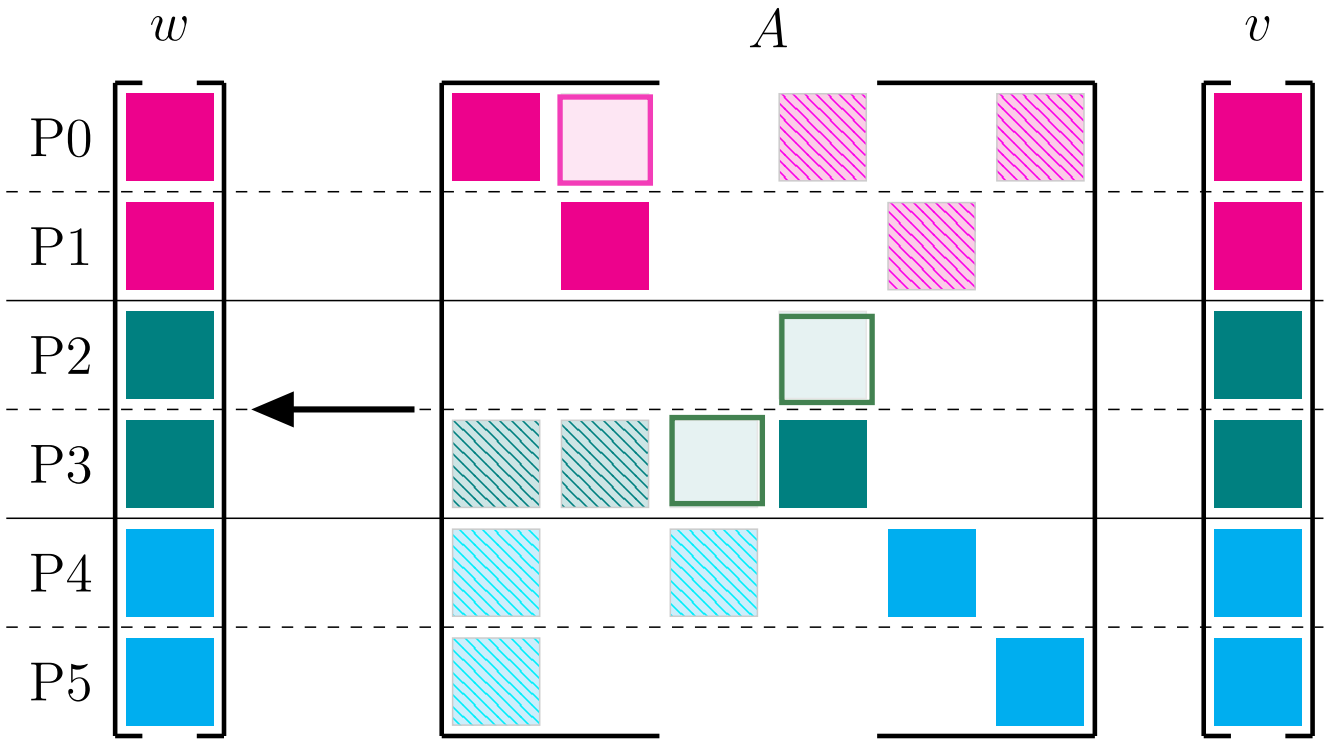
- Solid blocks : on-process
- Patterned blocks : off-process

Increased density → More patterned blocks → More communication

Node-Awareness

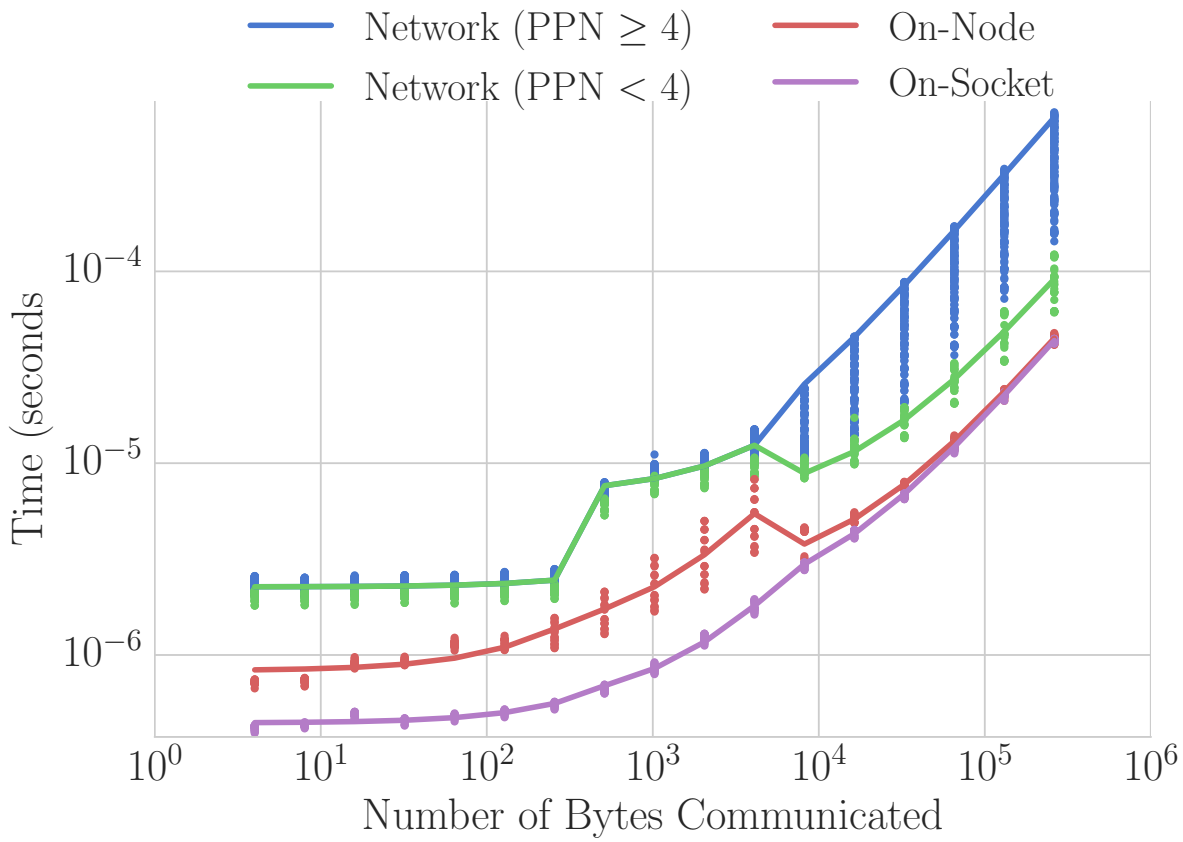


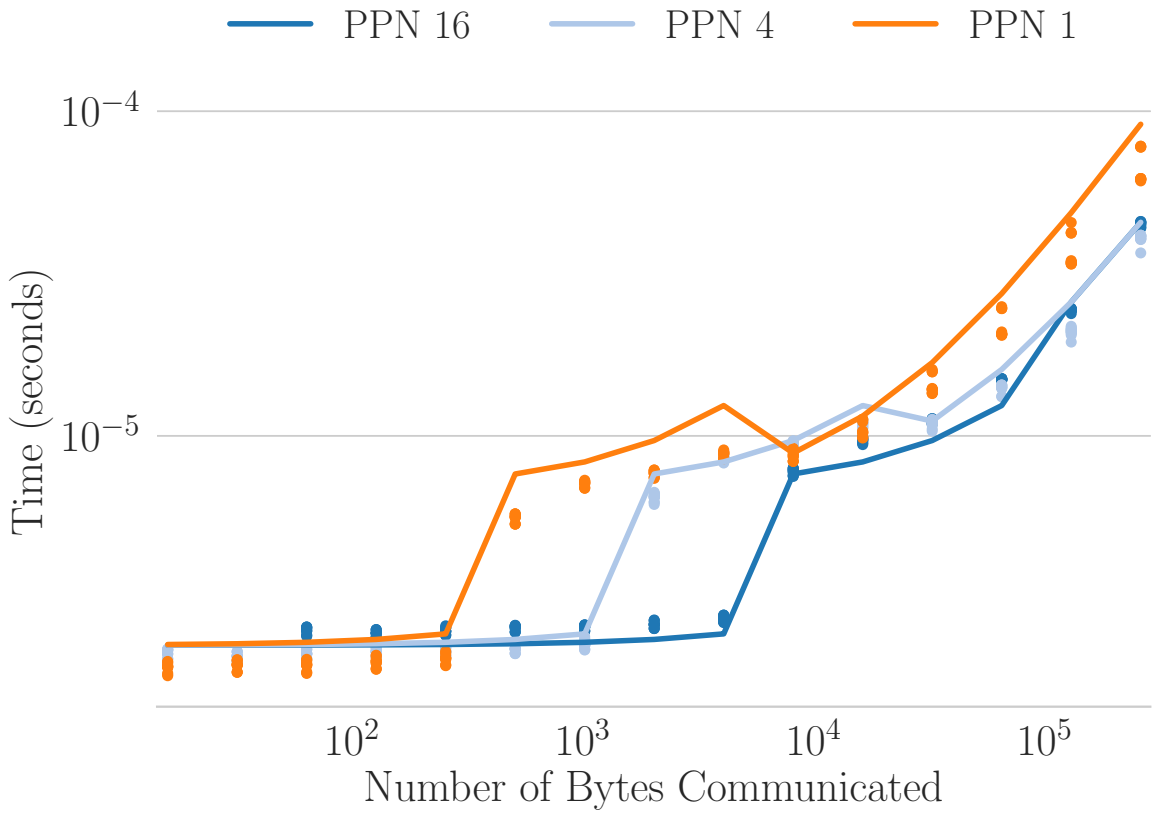
Six processes distributed across three nodes



Linear system distributed across the processes

Data Movement Costs



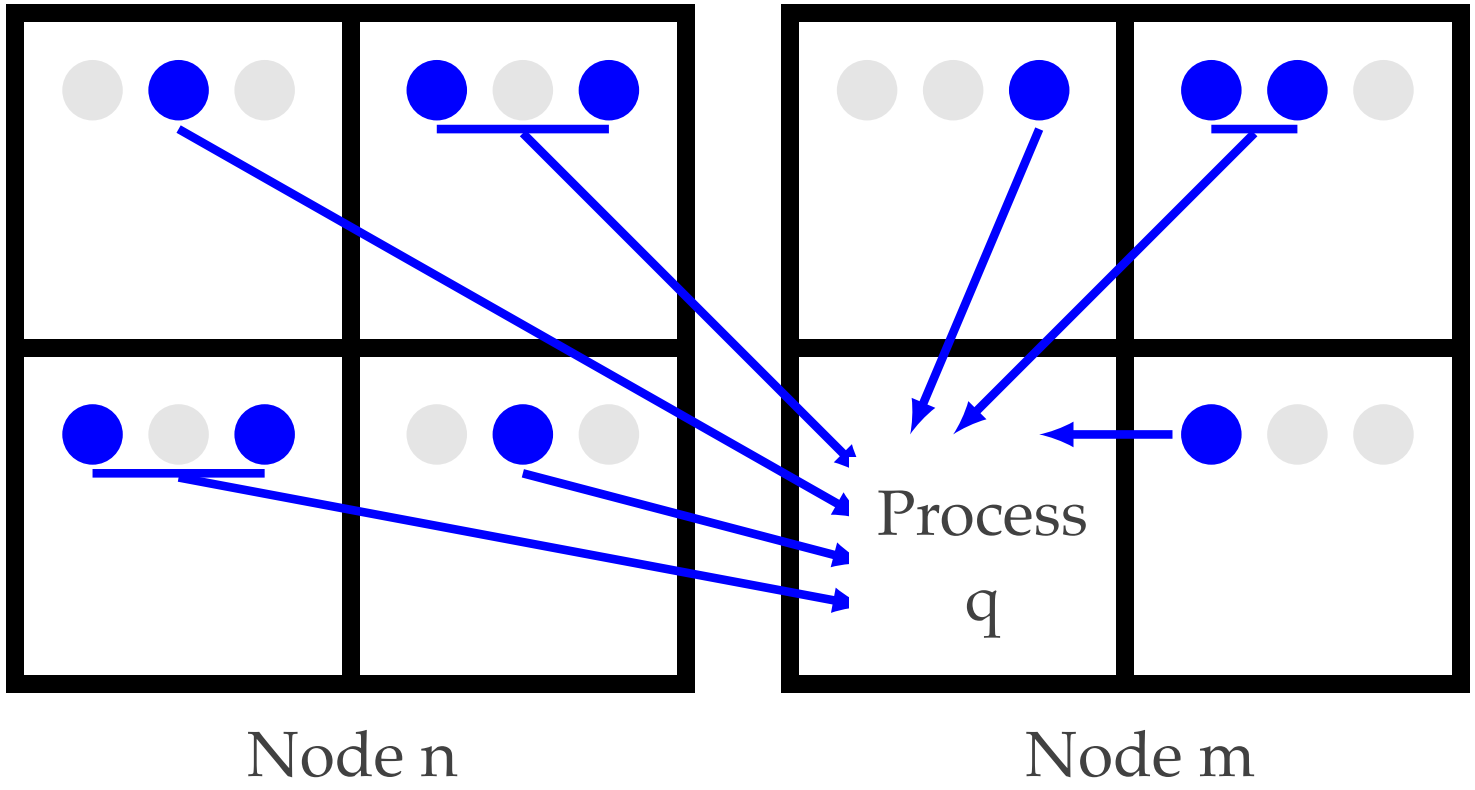


Locality-Aware Communication

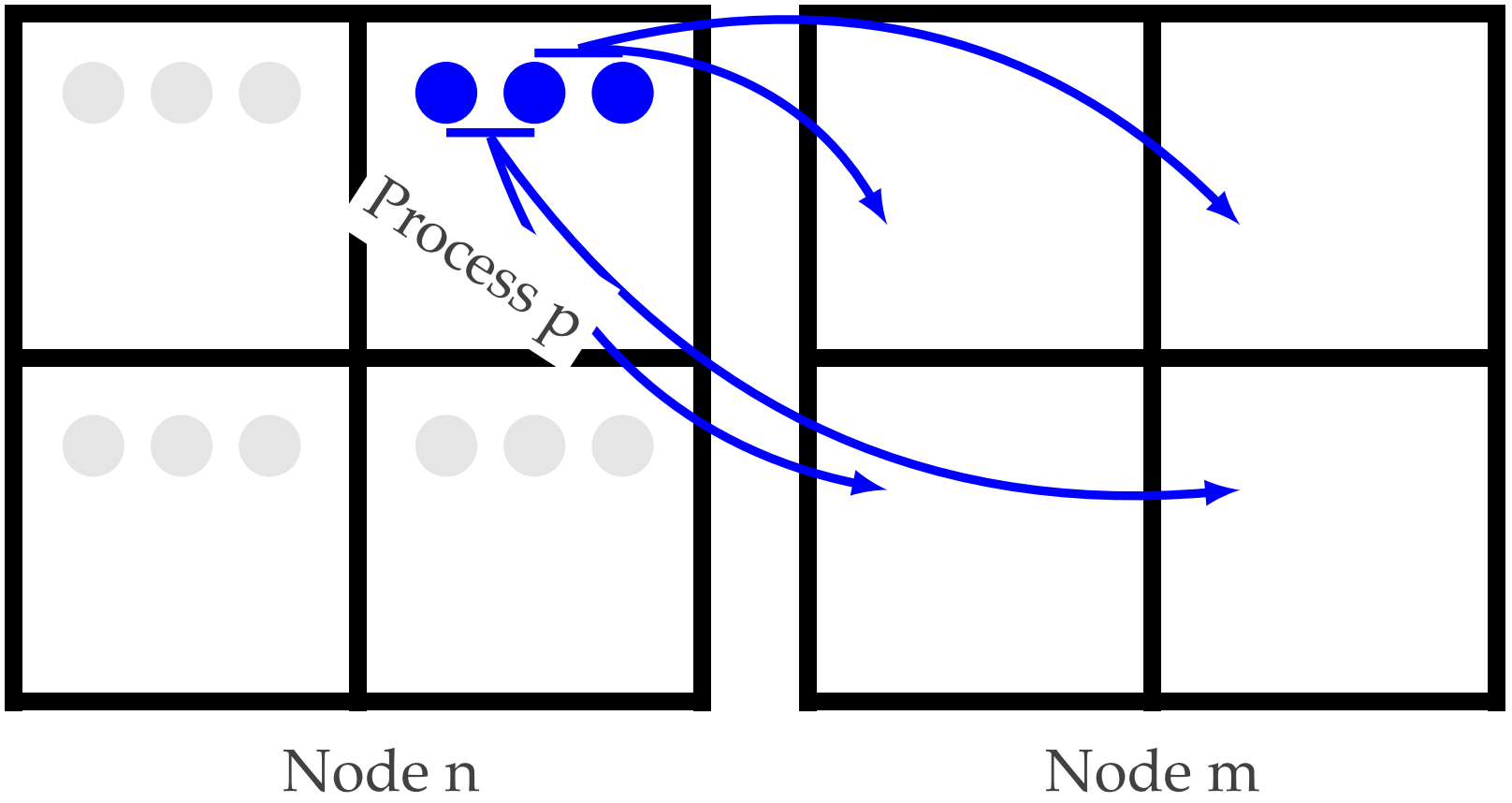
- Aggregate cheaper (e.g on-node) messages to reduce more expensive messages (e.g. off-node)

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

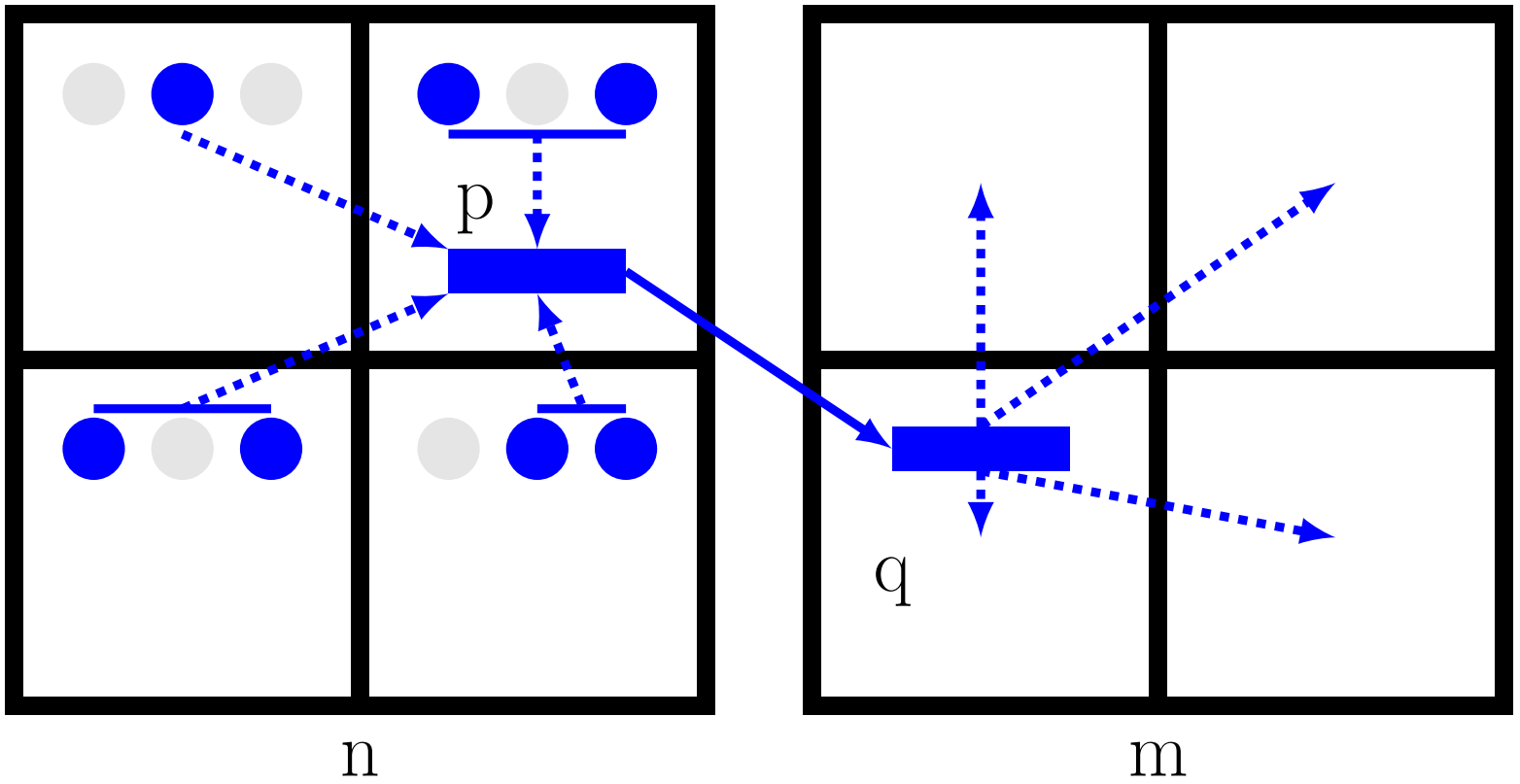
Standard Communication



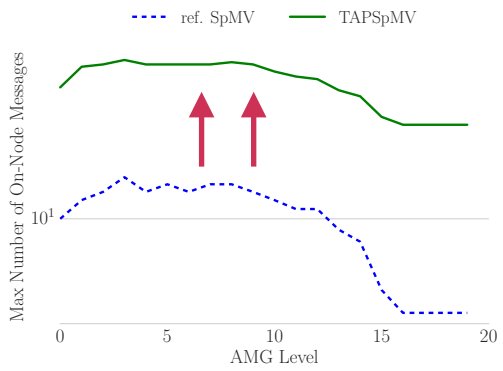
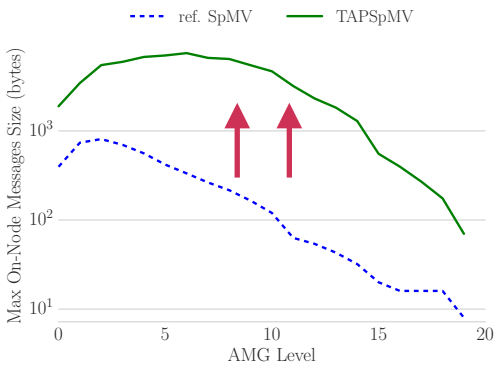
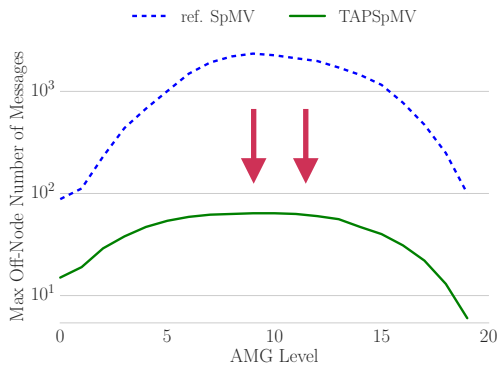
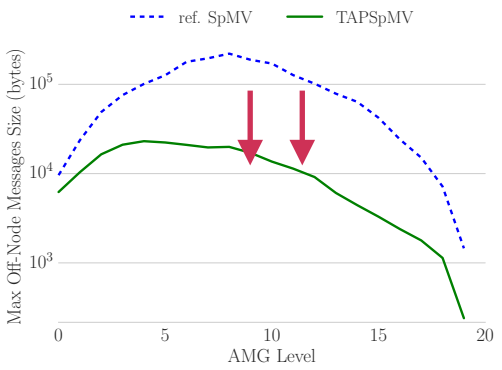
Standard Communication

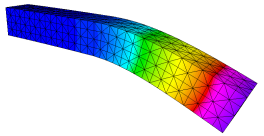


Locality-Aware Communication : Small Messages



Locality-Aware Communication

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

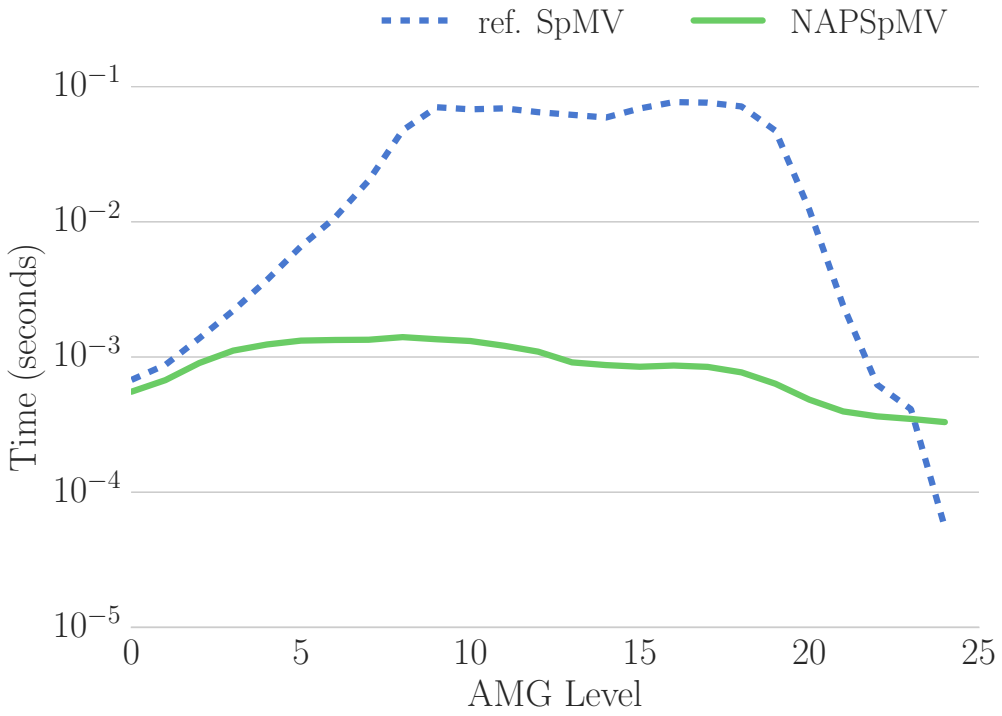


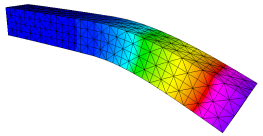
linear elasticity hierarchy
16,284 processes

Blue Dotted Lines :
Standard Communication

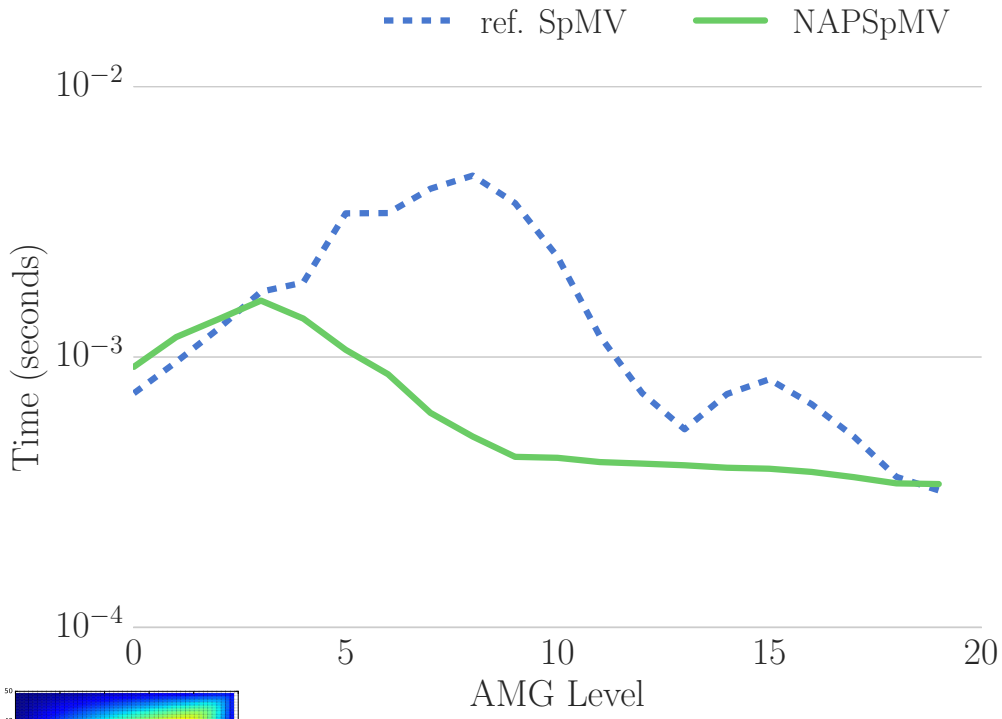
Green Lines:
Locality-Aware

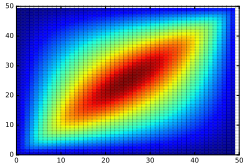
Locality-Aware SpMVs





Linear Elasticity (MFEM)

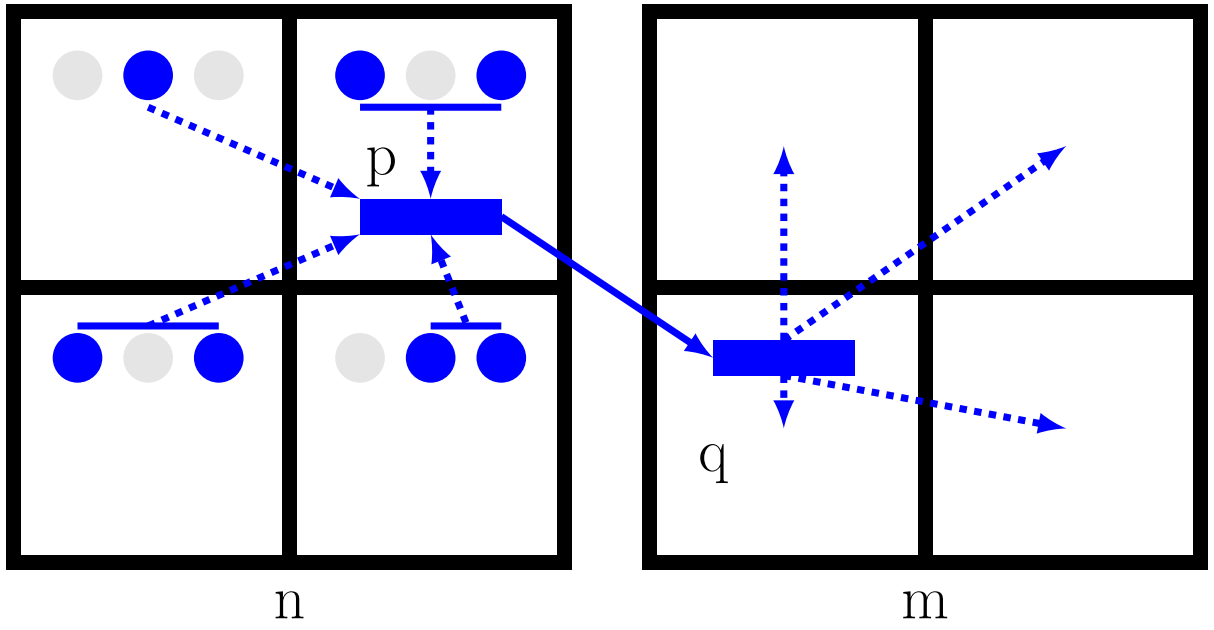


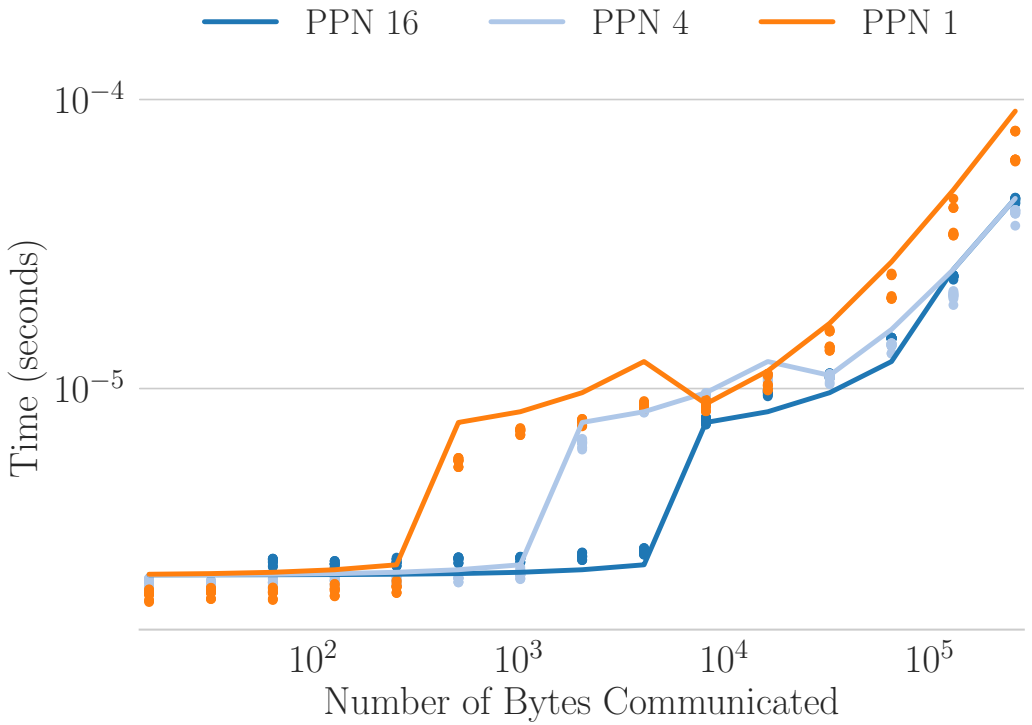


2D Rotated Anisotropic Diffusion

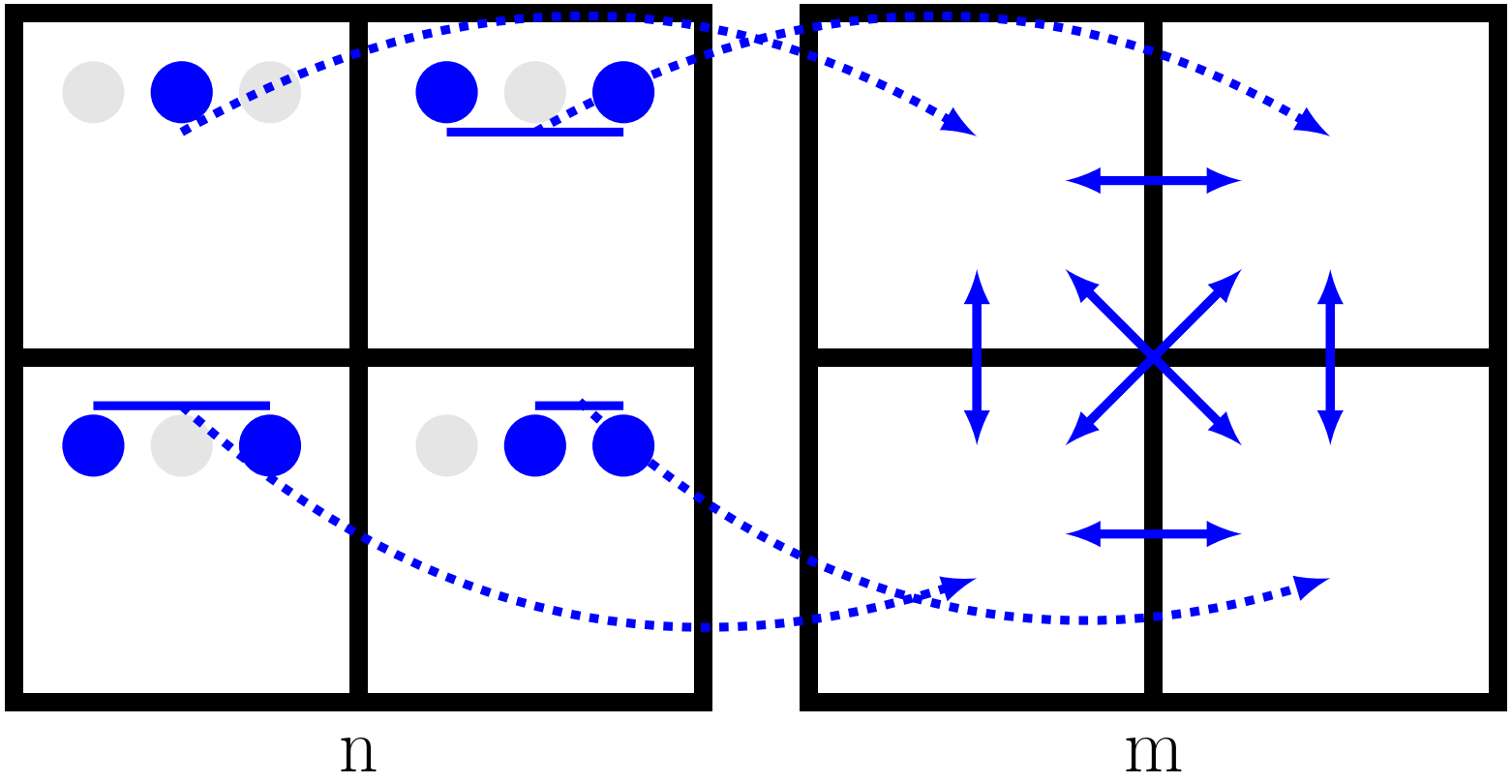
Node-Aware Sparse Matrix Vector Multiplication (https://doi.org/10.1016/j.jpdc.2019.03.016)

Locality-Aware Communication : Small Messages

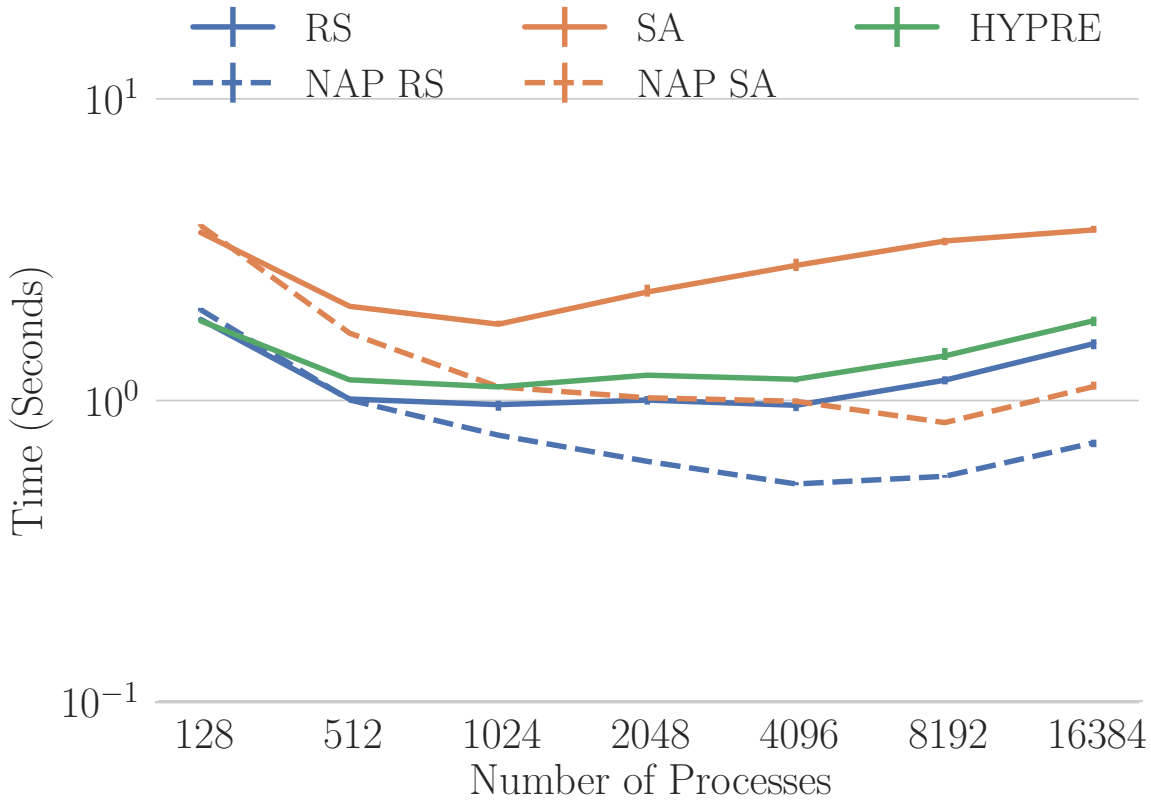




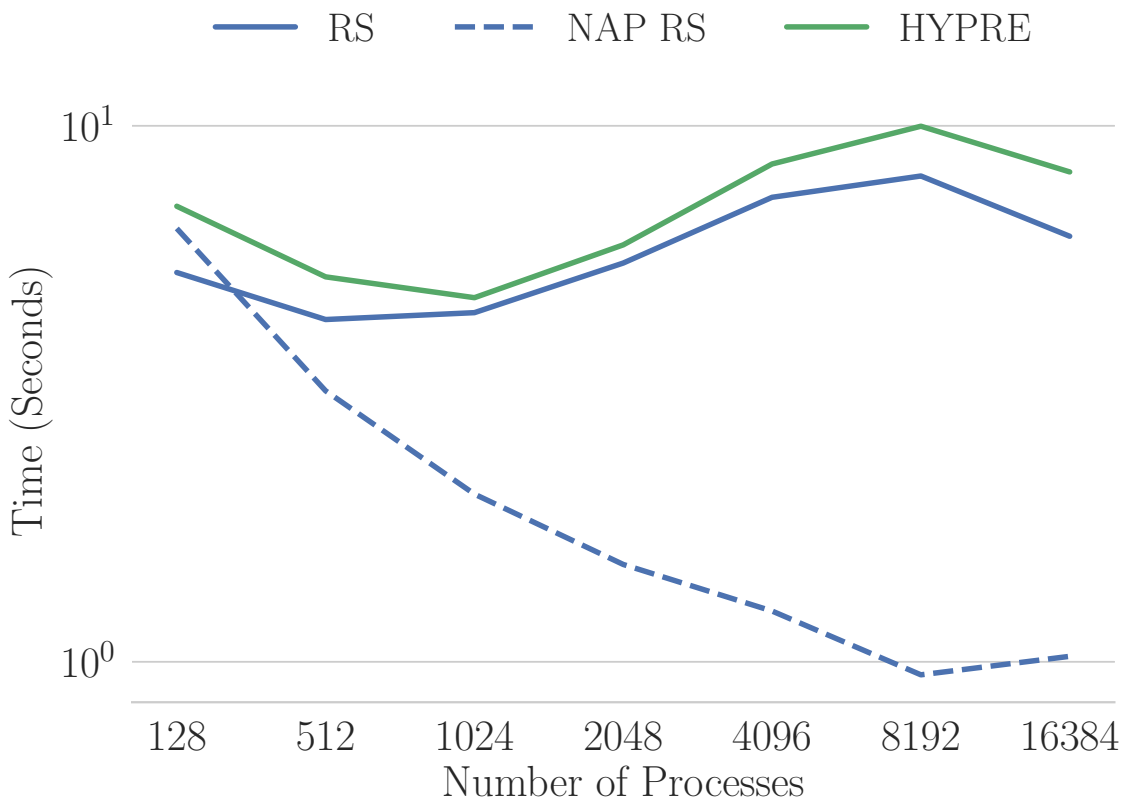
Locality-Aware Communication: Large Messages



Locality-Aware AMG

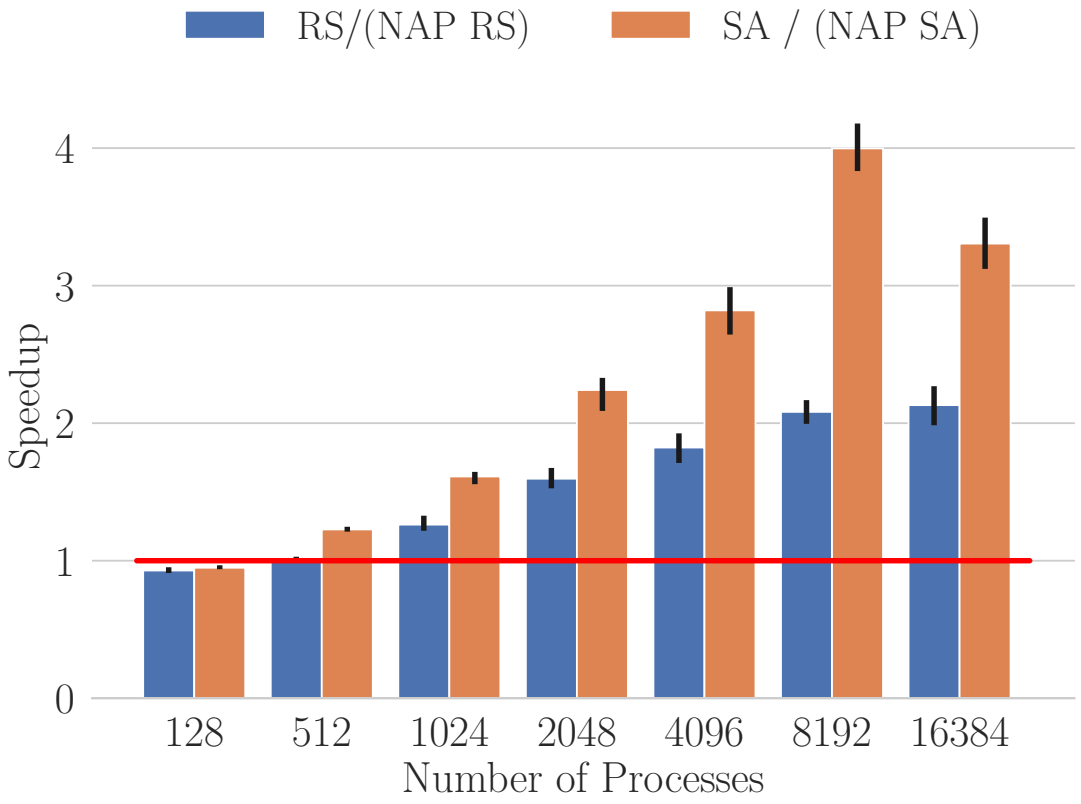


Grad-Div (MFEM)

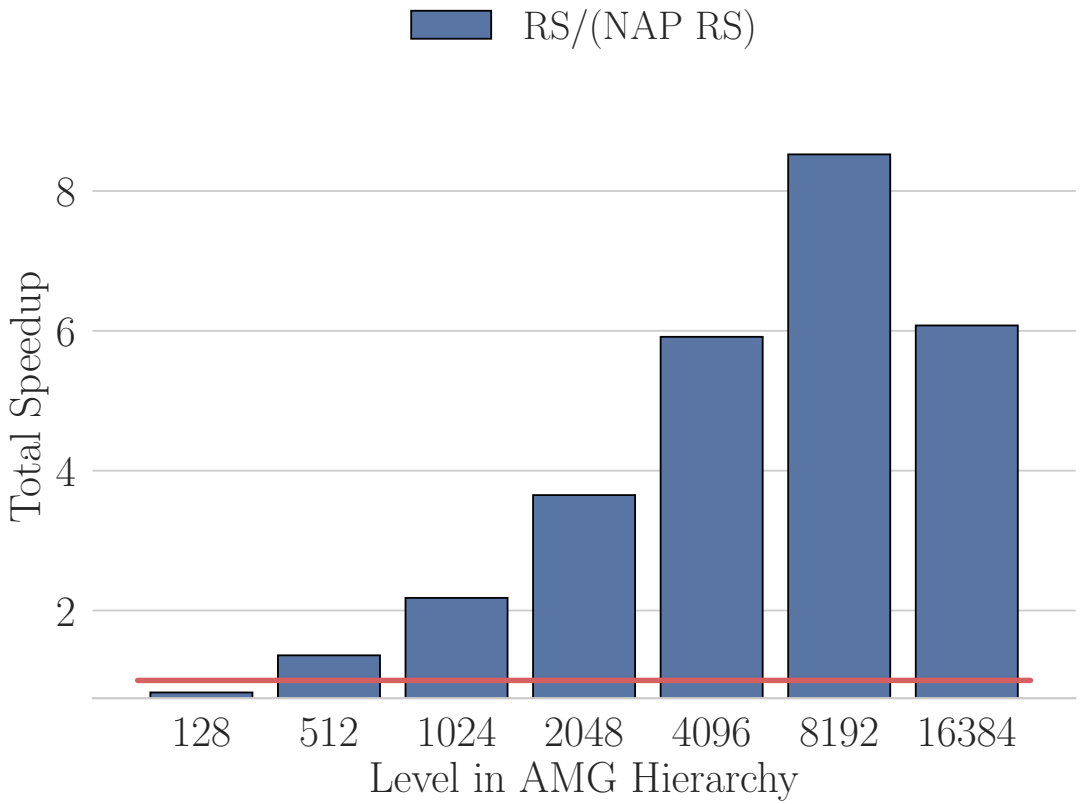


Laplacian (MFEM)

Locality-Aware AMG



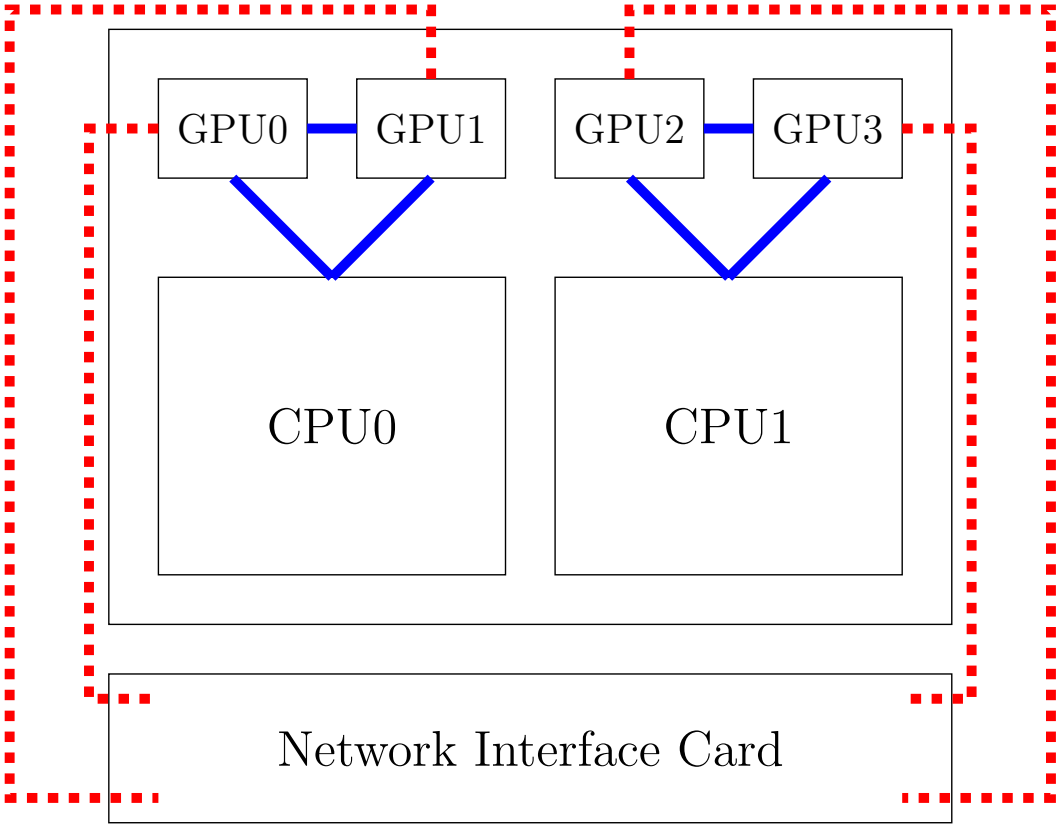
Grad-Div (MFEM)



Laplacian (MFEM)

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

Heterogeneous Architectures

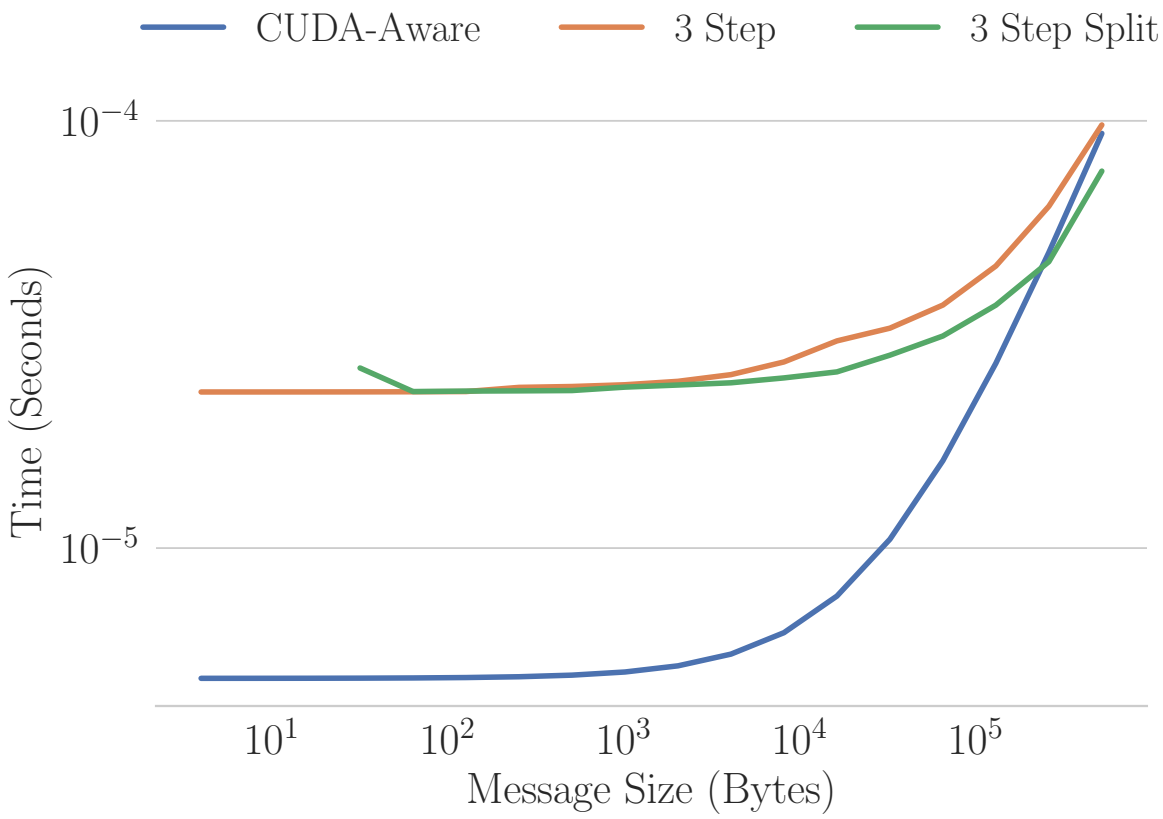


Summit (ORNL) and Lassen (LLNL)

What is the cheapest way to communicate data between GPUs?

1. CUDA-Aware + GPU Direct
2. Copy to CPU
3. Copy to many CPUs

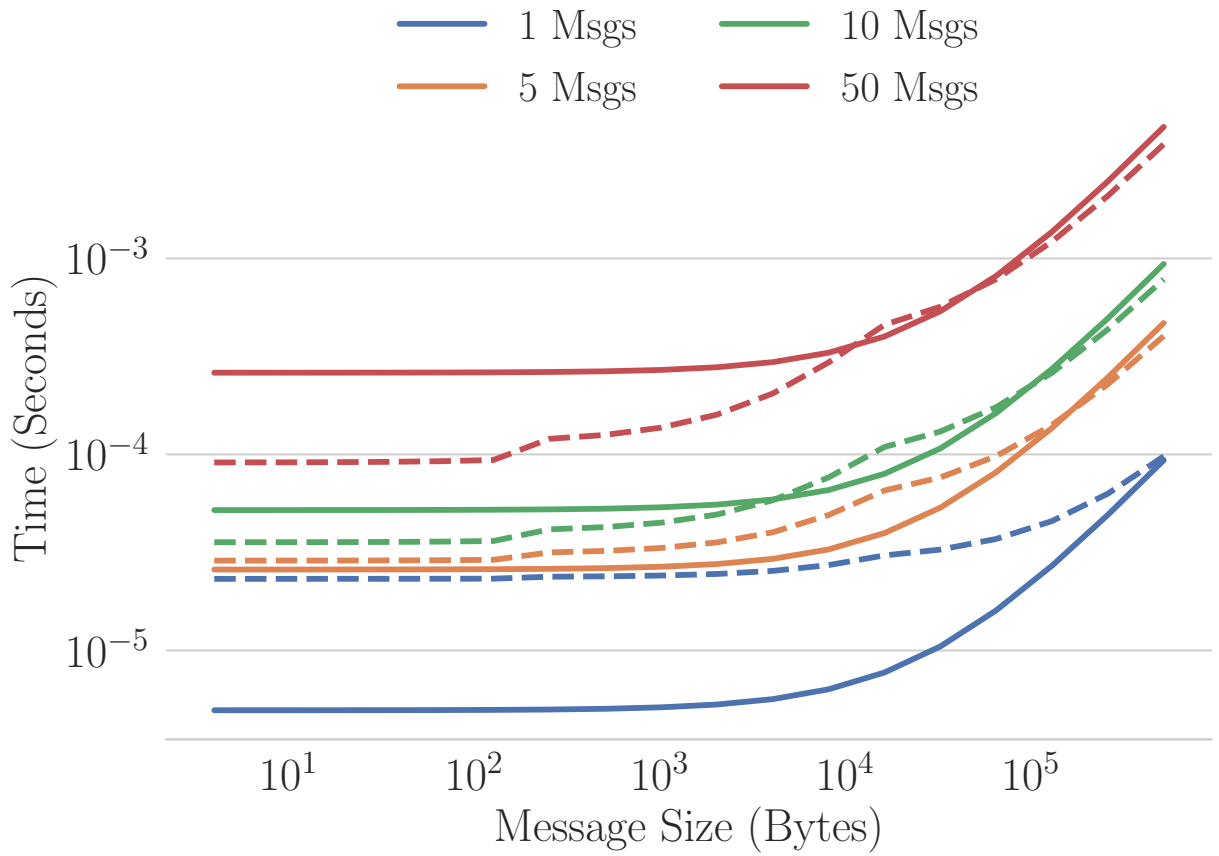
Communicating a Single Message



If messages are relatively small, it is always cheaper to send a single message with GPUDirect

When copying to CPU, typically slightly cheaper to split data across all available CPU cores

Communicating Multiple Messages

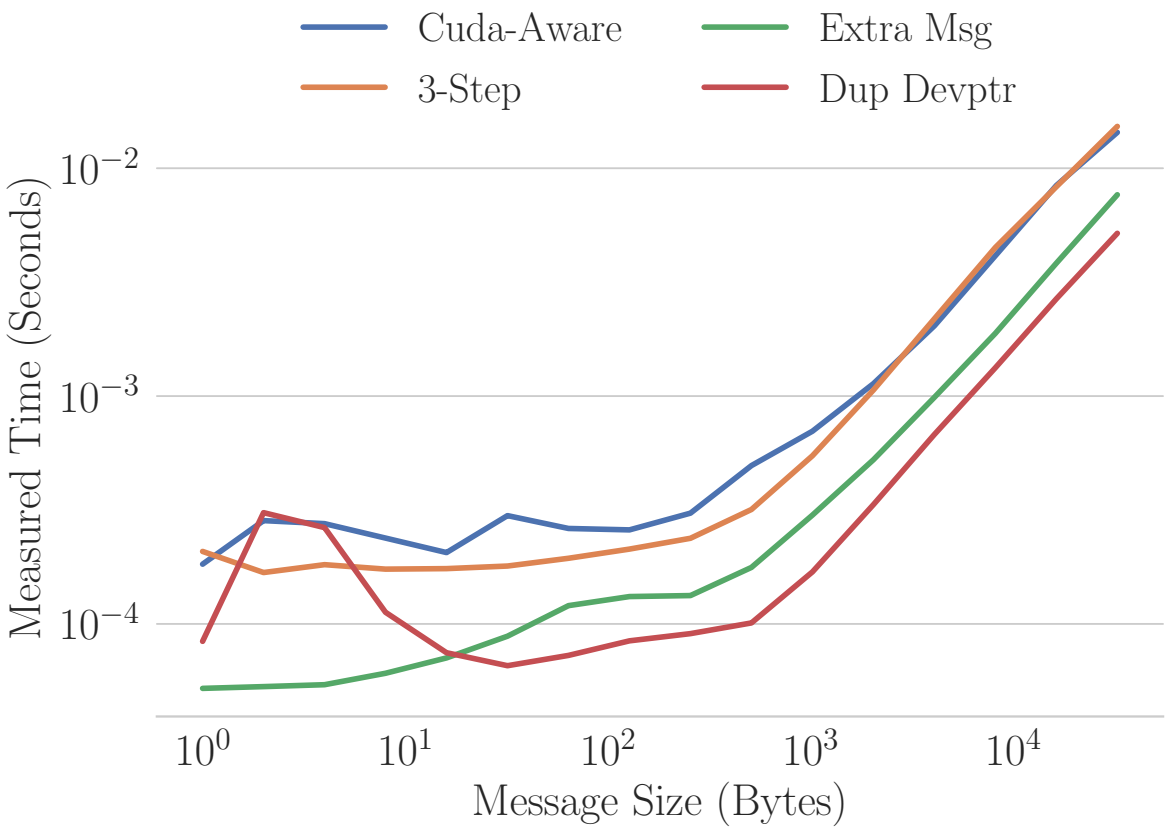


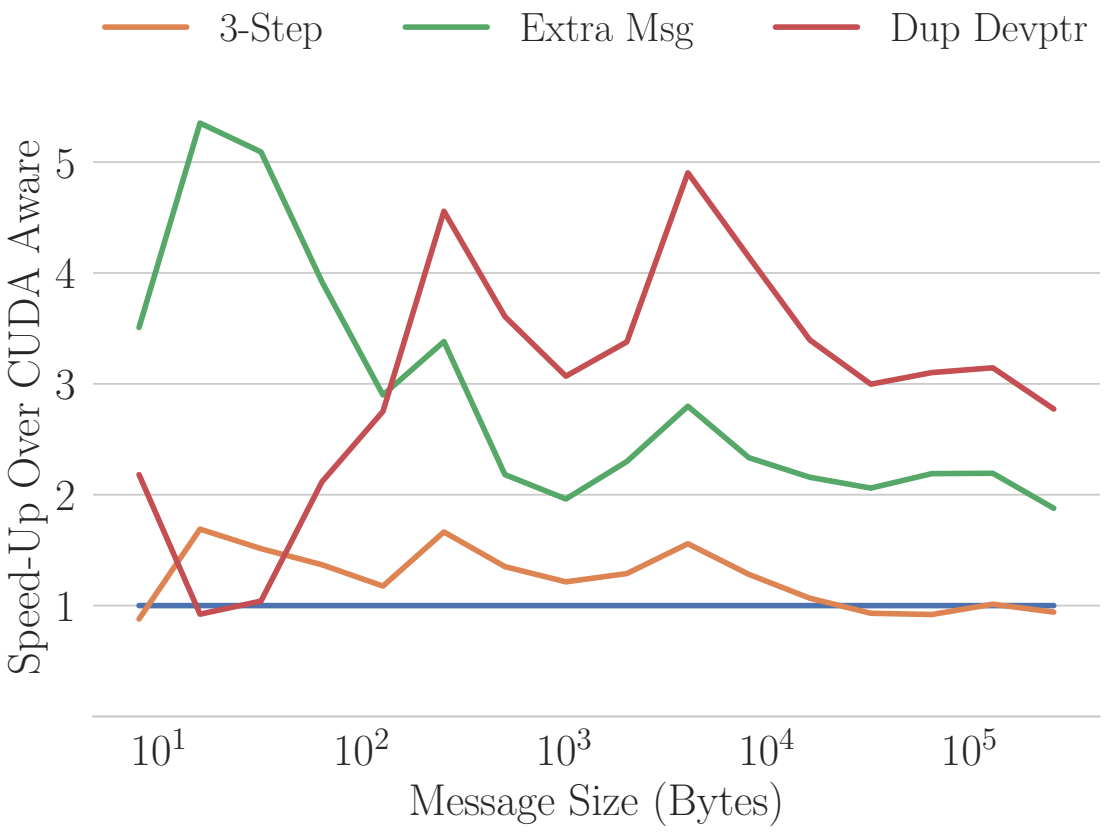
Solid Lines : CUDA-Aware , GPUDirect
Dotted Lines : Copy to CPU

When sending multiple messages, only need to copy to CPU once

If sending 10 or more messages, cheapest to copy to CPU and communicate between CPUs

MPI_Alltoallv Performance : 32 Nodes





Modeling Data Movement Performance on Heterogeneous Architectures (https://arxiv.org/abs/2010.10378)

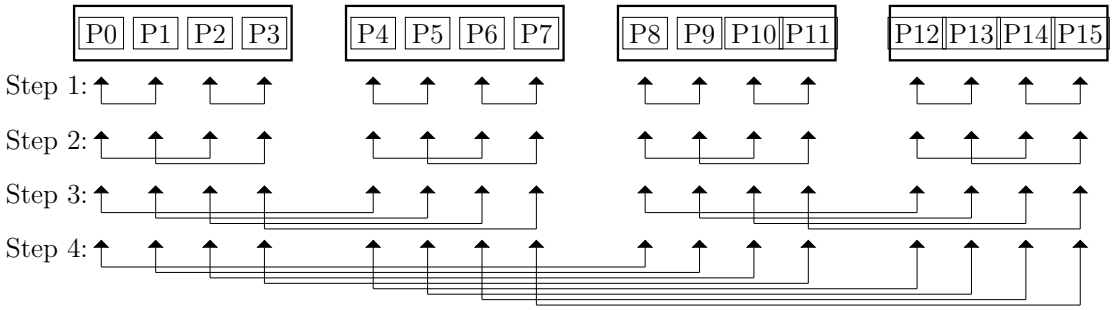
Applying Ideas to Other Methods

- Locality-awareness can be used to optimize collective operations
- On heterogeneous architectures, collective operations can be improved by utilizing all available CPU cores
- Neighborhood collectives are a natural fit for locality-aware methods

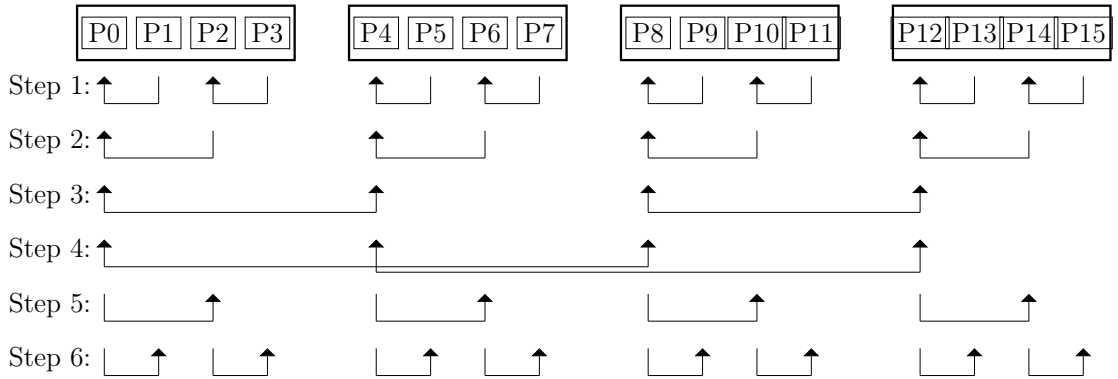
Locality-Aware MPI_Allreduce

- Natural idea for collectives (of small size):
  - Fewer active processes means fewer messages
- However, we want to minimize the number of inter-node messages, but care less about reducing intra-node communication
- Can improve performance of the Allreduce by having each process on a node exchange data with a different node at each step

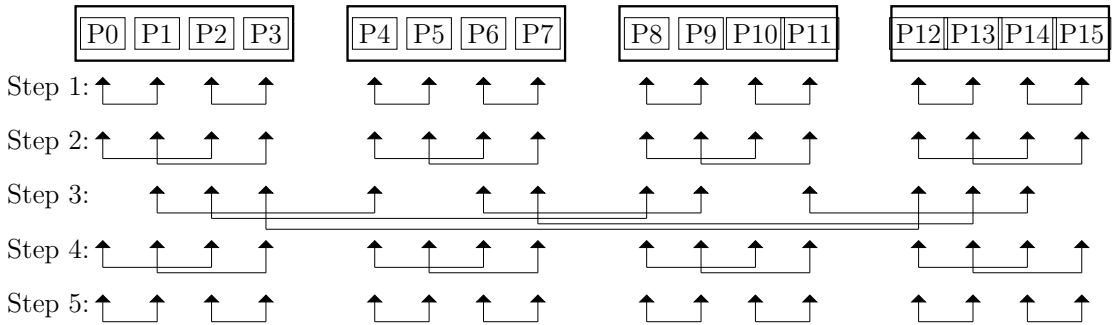
Locality-Aware MPI_Allreduce



Recursive Doubling

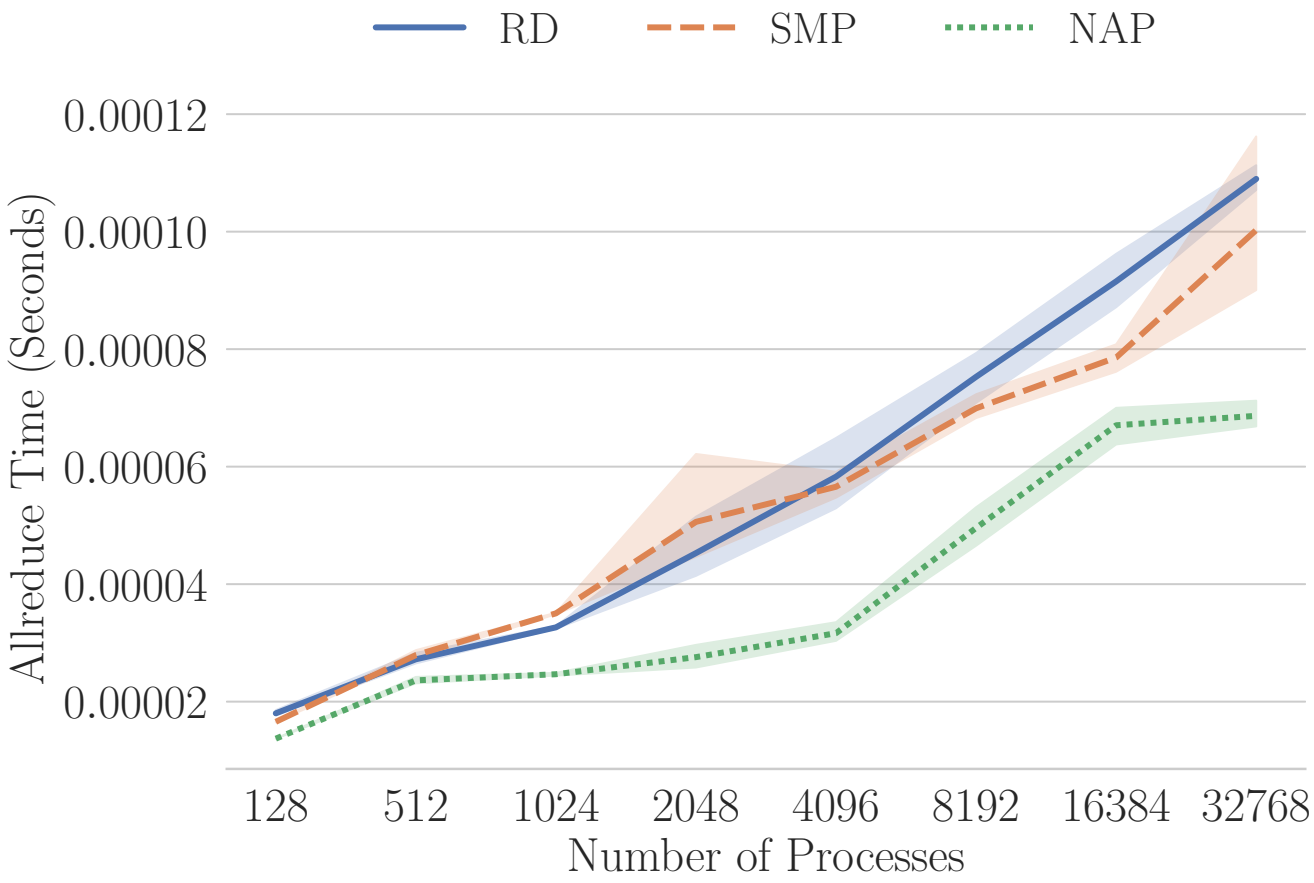


SMP (MPICH)



Locality-Aware

Locality-Aware MPI_Allreduce



Cost of reducing a single double per process

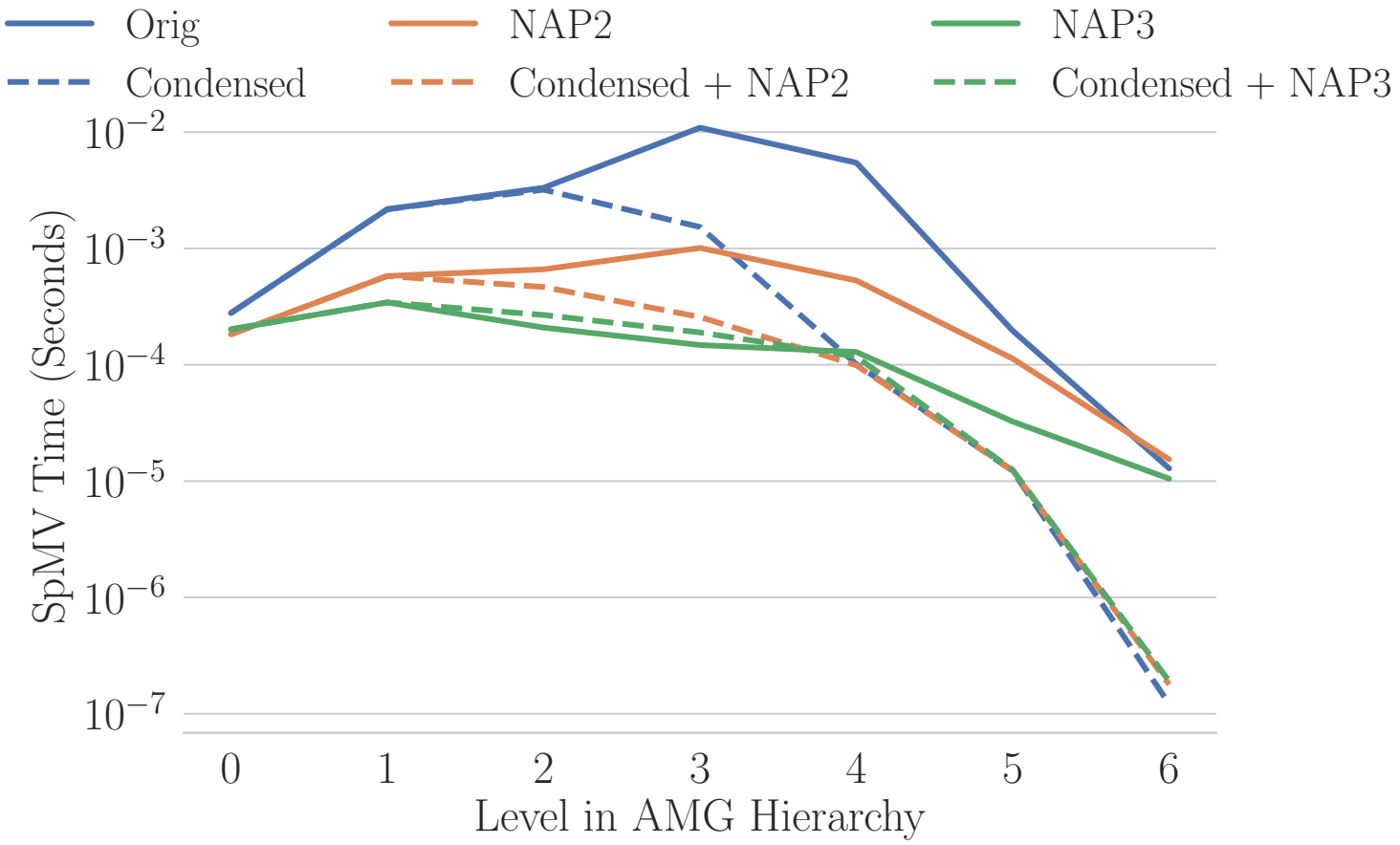
Blue : Recursive Doubling
Orange : SMP (master process per node)
Green : Locality-Aware

Node-Aware Improvements to Allreduce (https://doi.org/10.1109/ExaMPI49596.2019.00008)

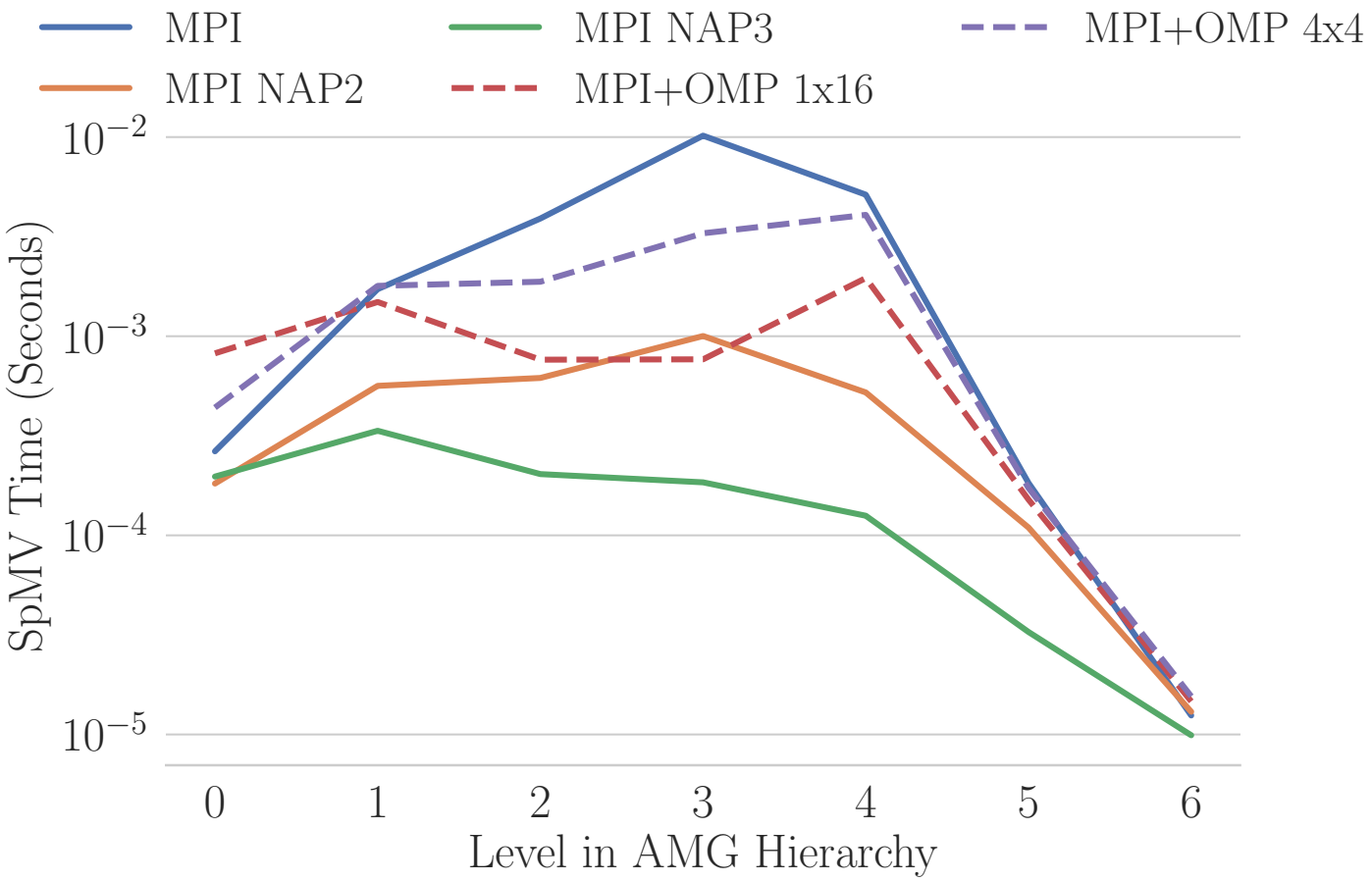
Thanks for your time!

Questions?

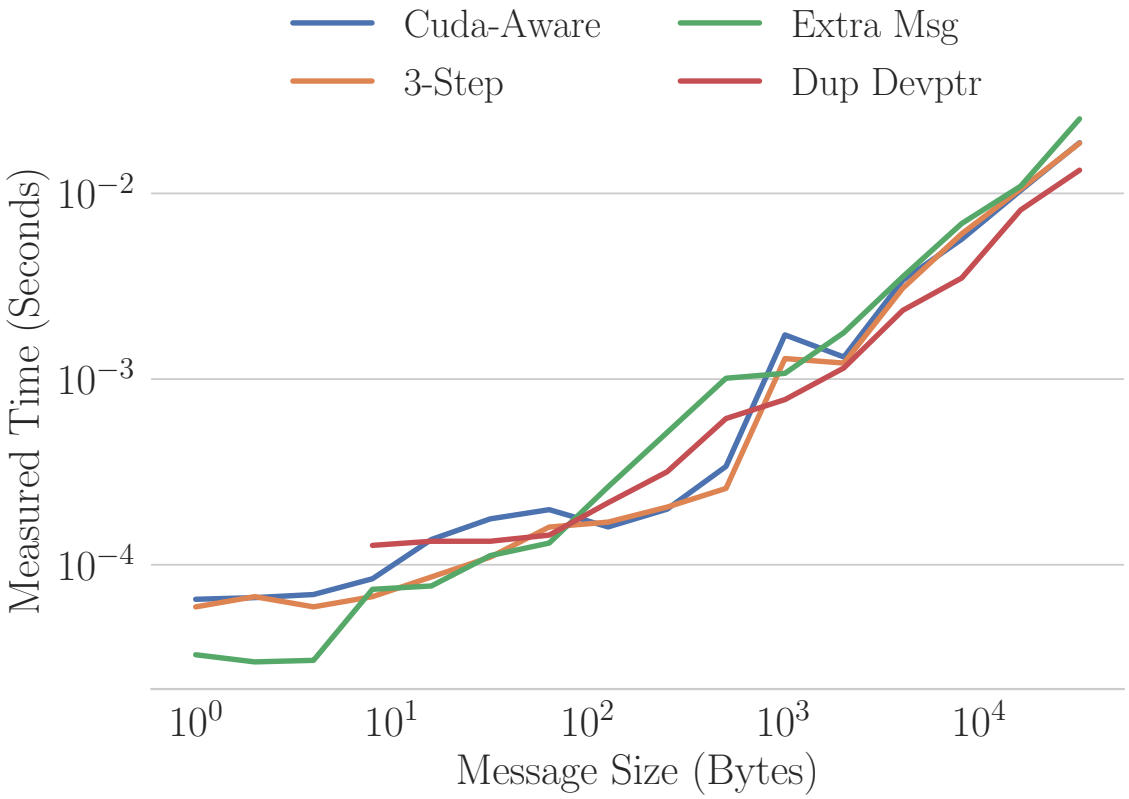
AMG - Condensed Coarse Grids

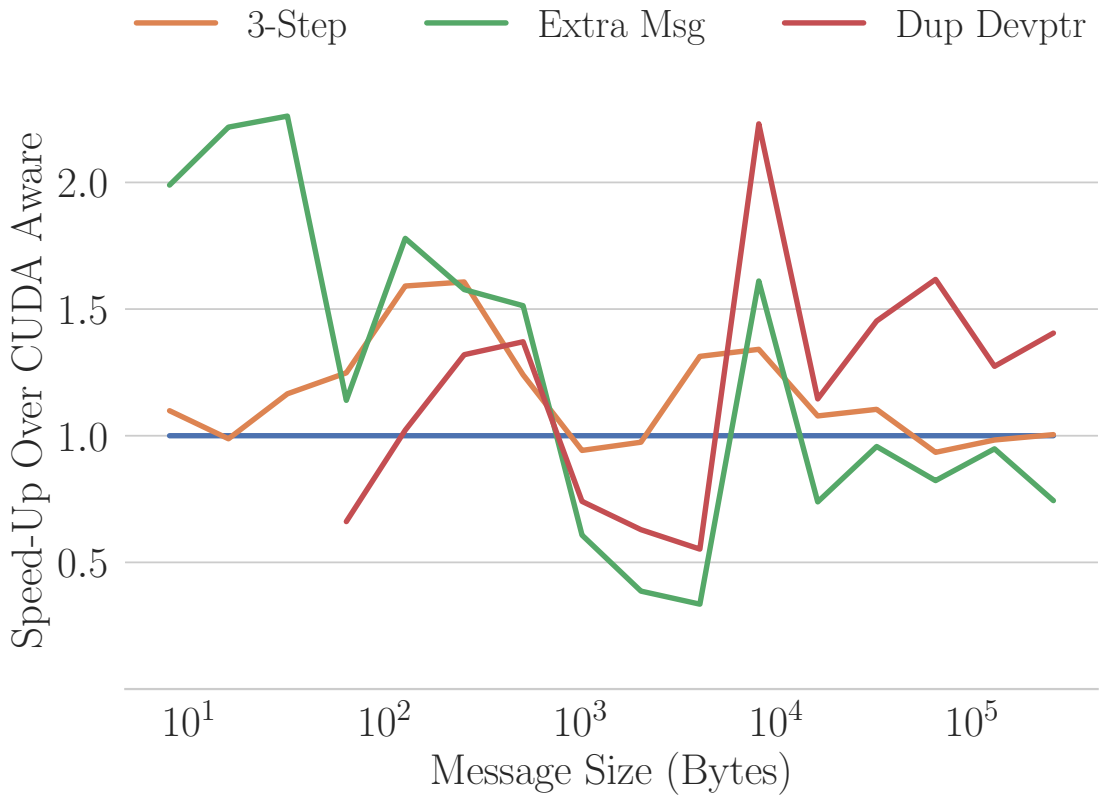


AMG SpMVs - OpenMP

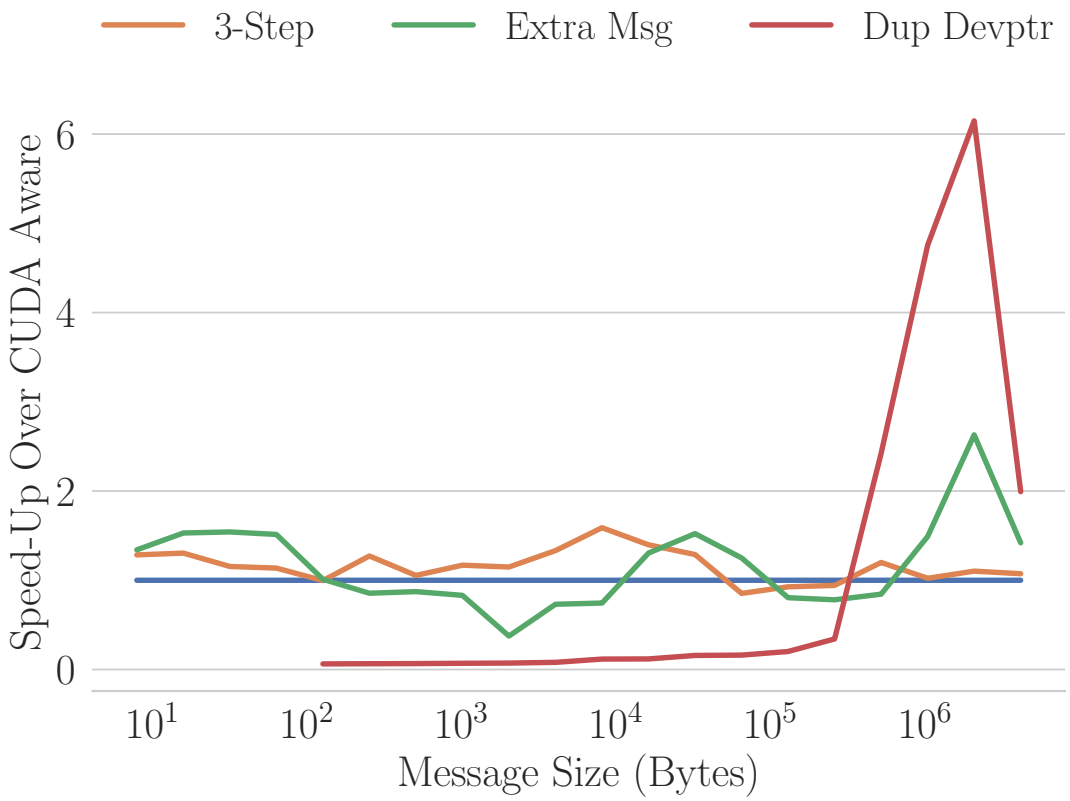


Summit MPI_Alltoall : 32 Nodes





Lassen MPI_Alltoall : 32 Nodes



Summit MPI_Allreduce : 32 Nodes