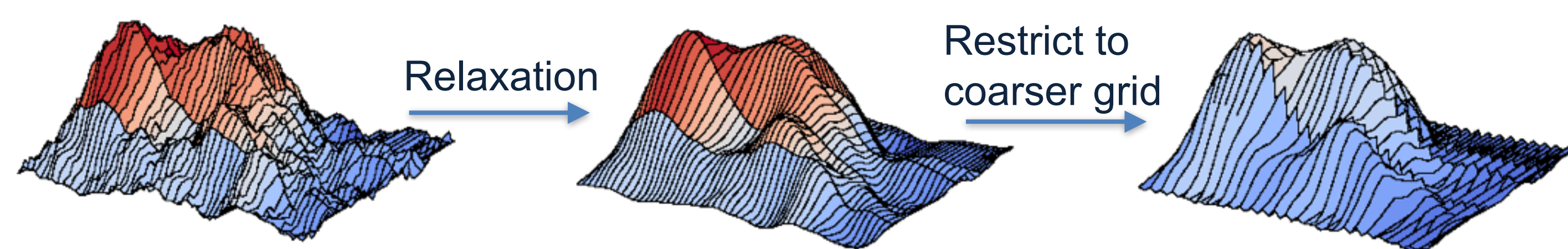
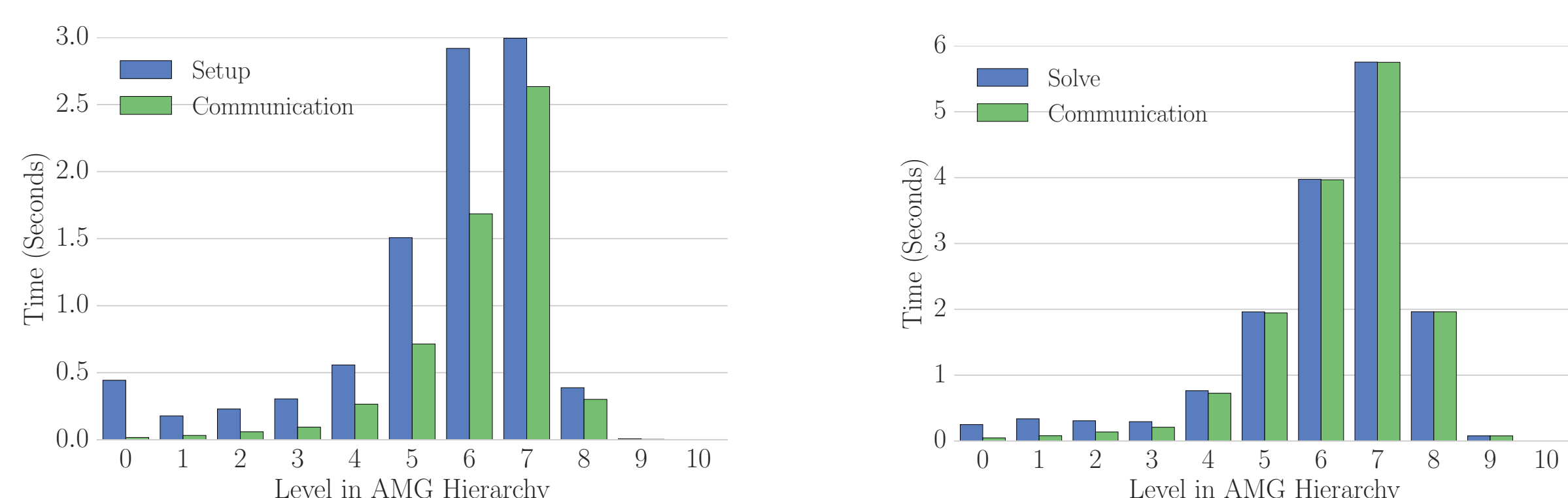


Algebraic Multigrid (AMG)

- Iterative method for sparse linear system



- Two phases: **setup** hierarchy and iteratively **solve**
- Dominant operations: triple sparse matrix produce (setup) and sparse matrix-vector multiplication (solve)
- Large communication costs on coarse levels**

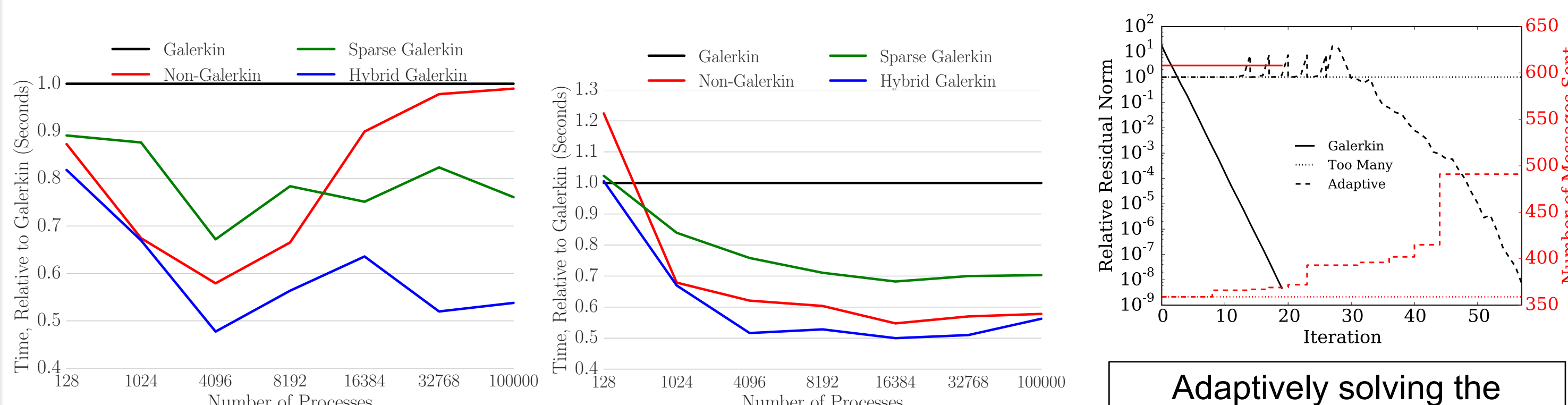
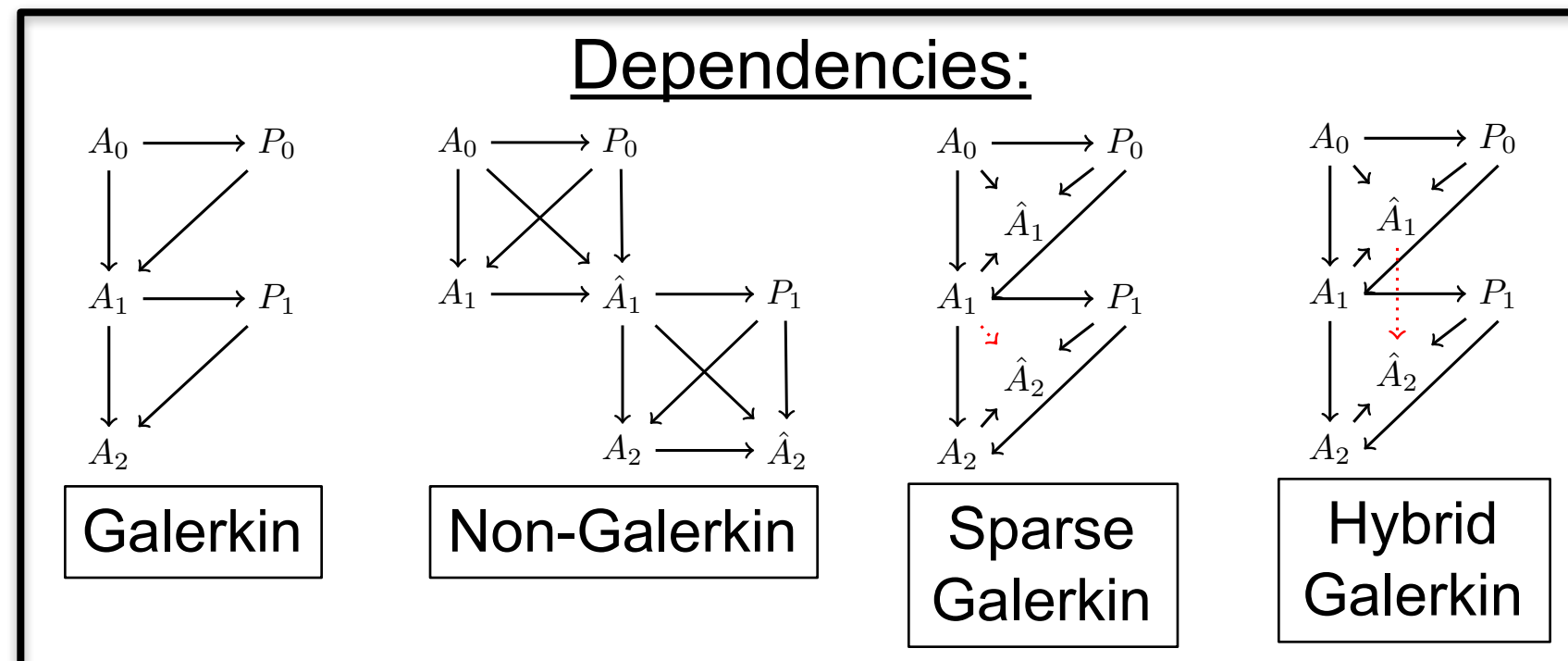


Altering Algorithm through Sparsification

- Main Idea: Remove small values from coarse levels**
- Based on method of non-Galerkin coarse grids
- Remove entries **after** hierarchy is formed

Sparsification Process:

- Form sparsity pattern: $M = P^T A \hat{P} + \hat{P}^T A P$ (\hat{P} is injection)
- For some γ , remove values not in M with relative magnitude less than γ .
- Optimal γ varies with system, scale, and level in hierarchy
- Small γ , costly communication; Large γ , divergence.



Time, relative to Galerkin AMG, to solve weakly (left) and strongly (right) scaled 2D rotated anisotropic diffusion

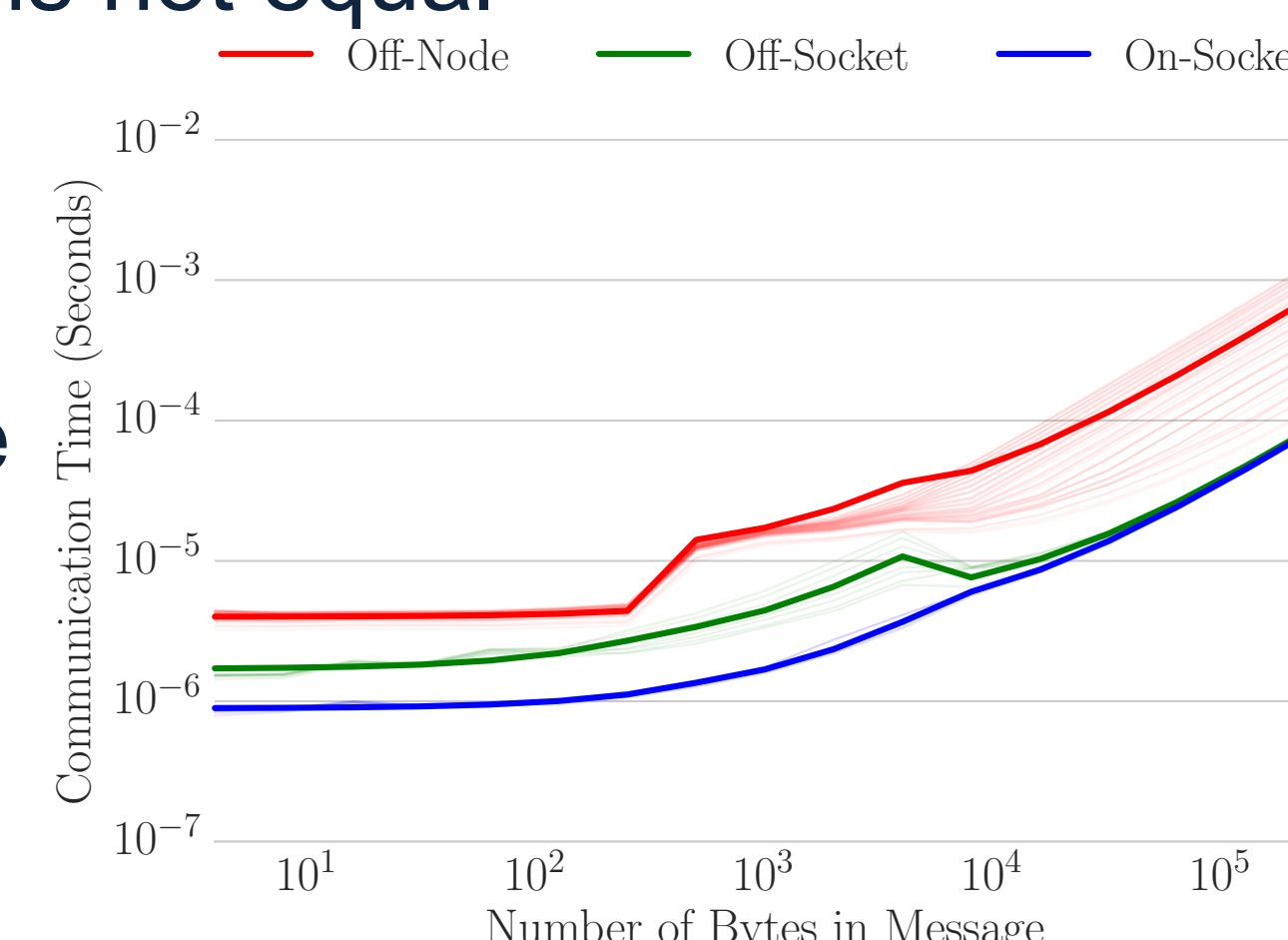
Adaptively solving the hierarchy, adding back entries to improve convergence as necessary

- Reduces cost of AMG (both weak and strong scaling)**
- If poor initial γ , hierarchy can be adapted during solve**

Altering Parallel Implementation with Topology-Aware Methods

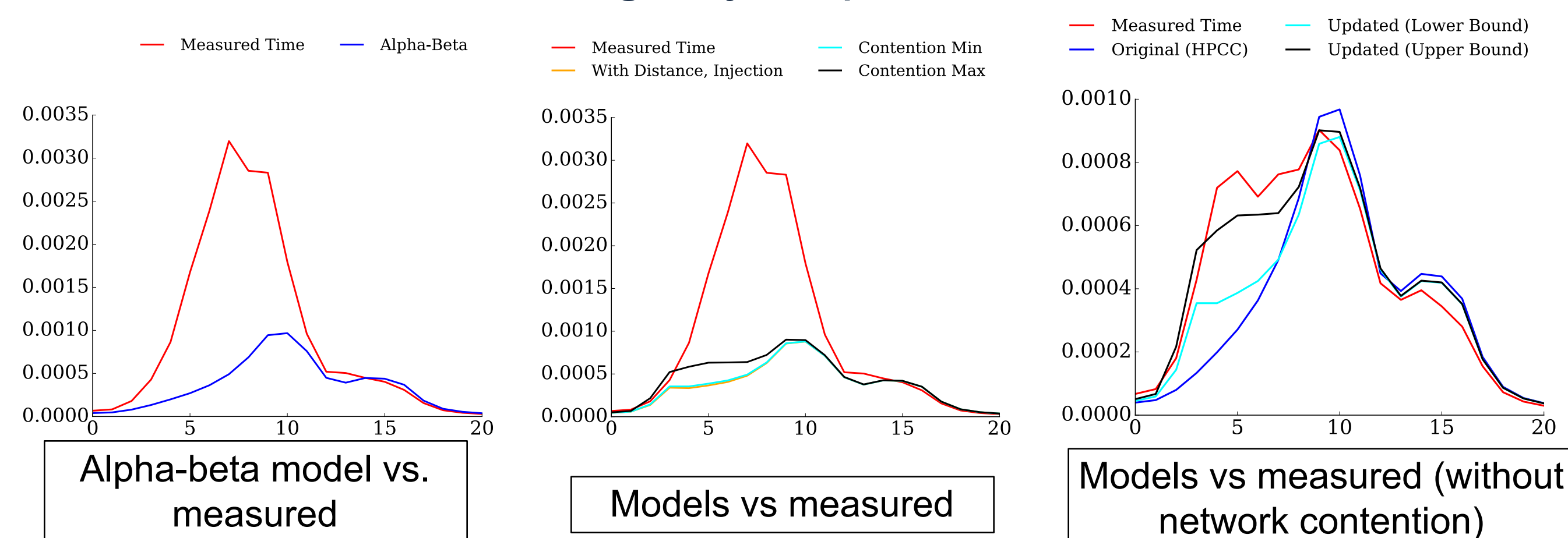
Performance Models

- Model costs: all communication is not equal
- Cost factors:
 - Size
 - Send / Receive location
- Improve models with knowledge of topology
- Separate alphas / betas for:
 - On socket
 - On node, off socket
 - Off node



Alpha-Beta model (thick) and timings (thin)

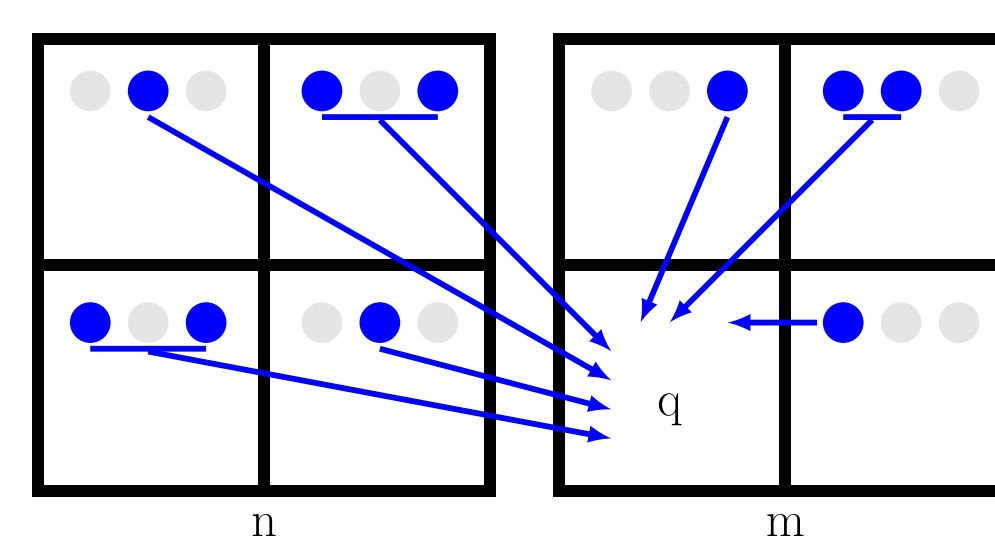
- Model penalties: traverse multiple links, injection bandwidth limits, maximum / average bytes per link



- In progress: max-rate model parameters, network contention measure, and queue search time*

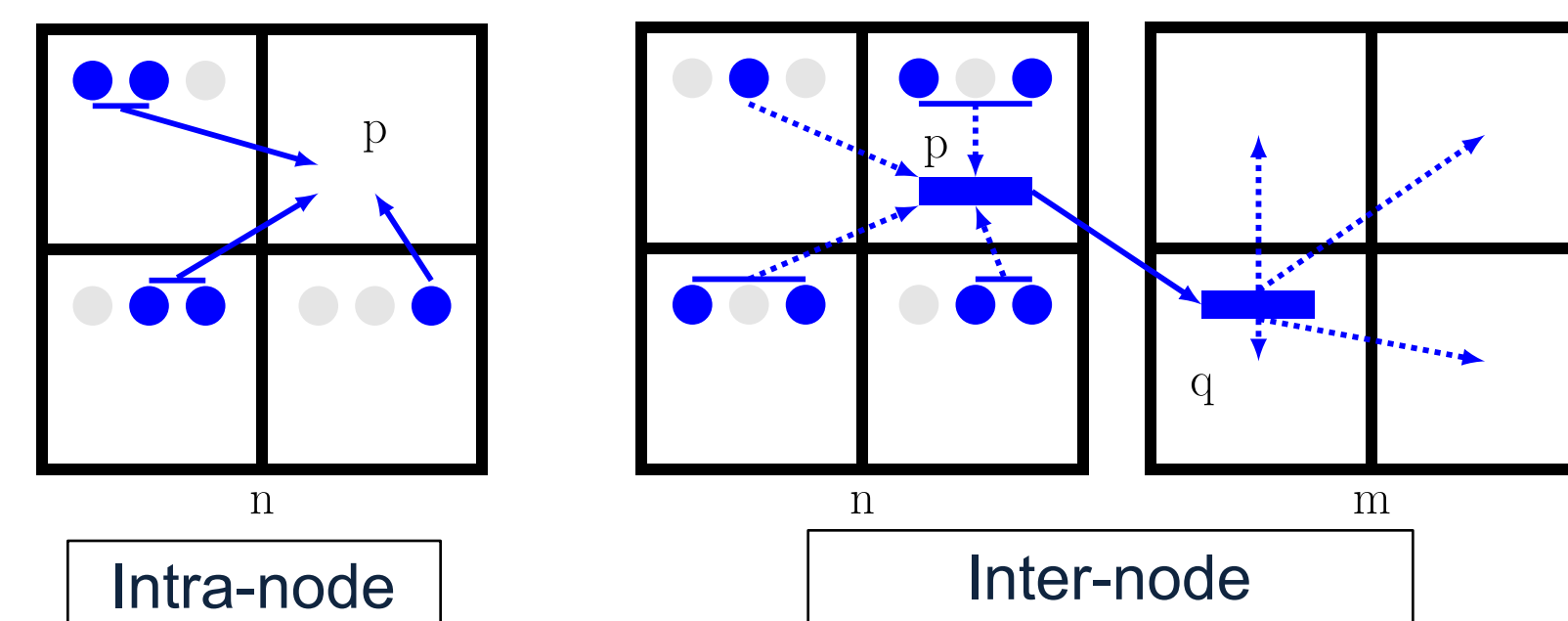
Node-Aware Communication

Standard



- Modern parallel computers have:
 - Large number of nodes
 - Many processes per node
 - Send directly to any process

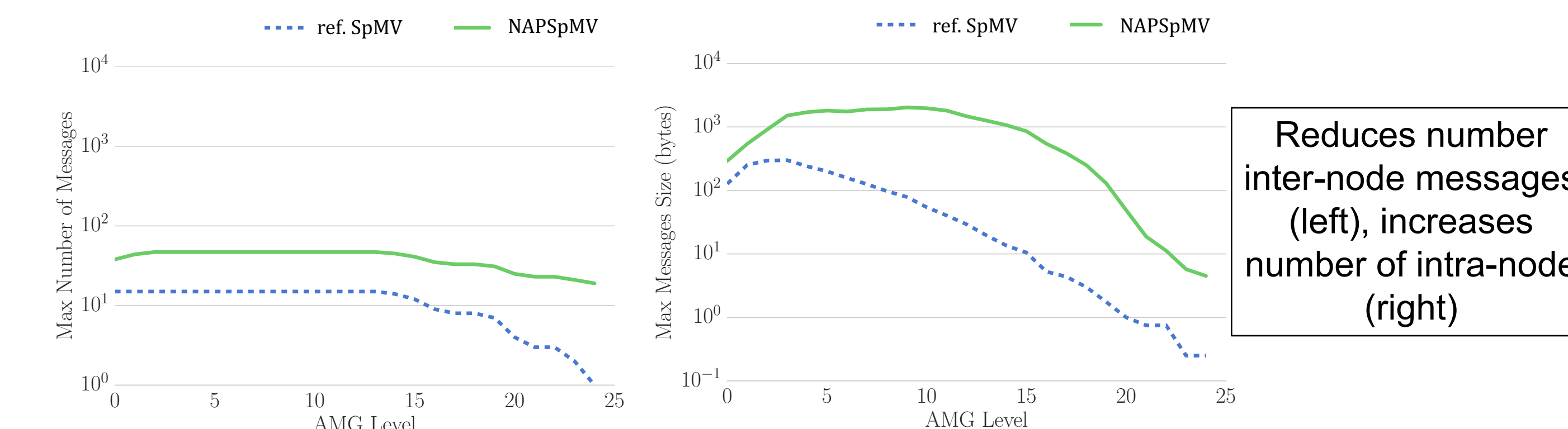
Node - Aware



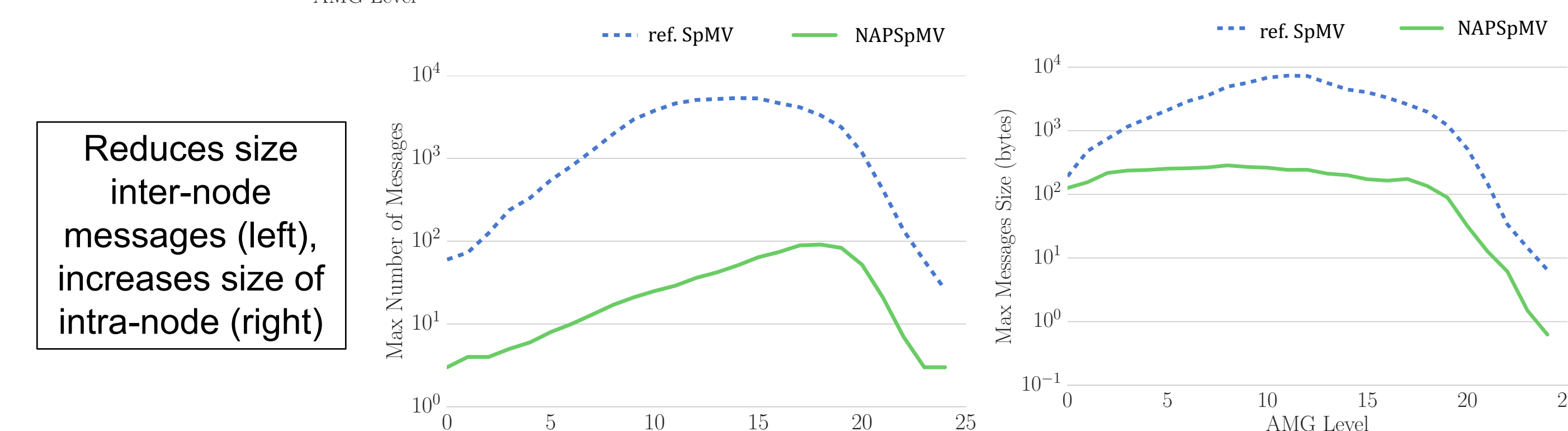
- Limit inter-node communication**
- Intra-node messages still sent directly
- 3 step inter-node
 - Send all data to node p
 - Single inter-node message from p to q
 - Redistribute data among node m
- Messages from rank on node n to rank on node m:
 - Send all data to node p
 - Single inter-node message from p to q
 - Redistribute data among node m
- Note: assumes many nodes in partition, so all local processes are sending and receiving, but only a single message between any two nodes*

Node Aware SpMV and SpGEMM

- MFEM linear elasticity, 16,384 processes of Blue Waters
- Node aware parallel (NAP) sparse matrix-matrix (SpGEMM) and sparse matrix-vector (SpMV) multiplication

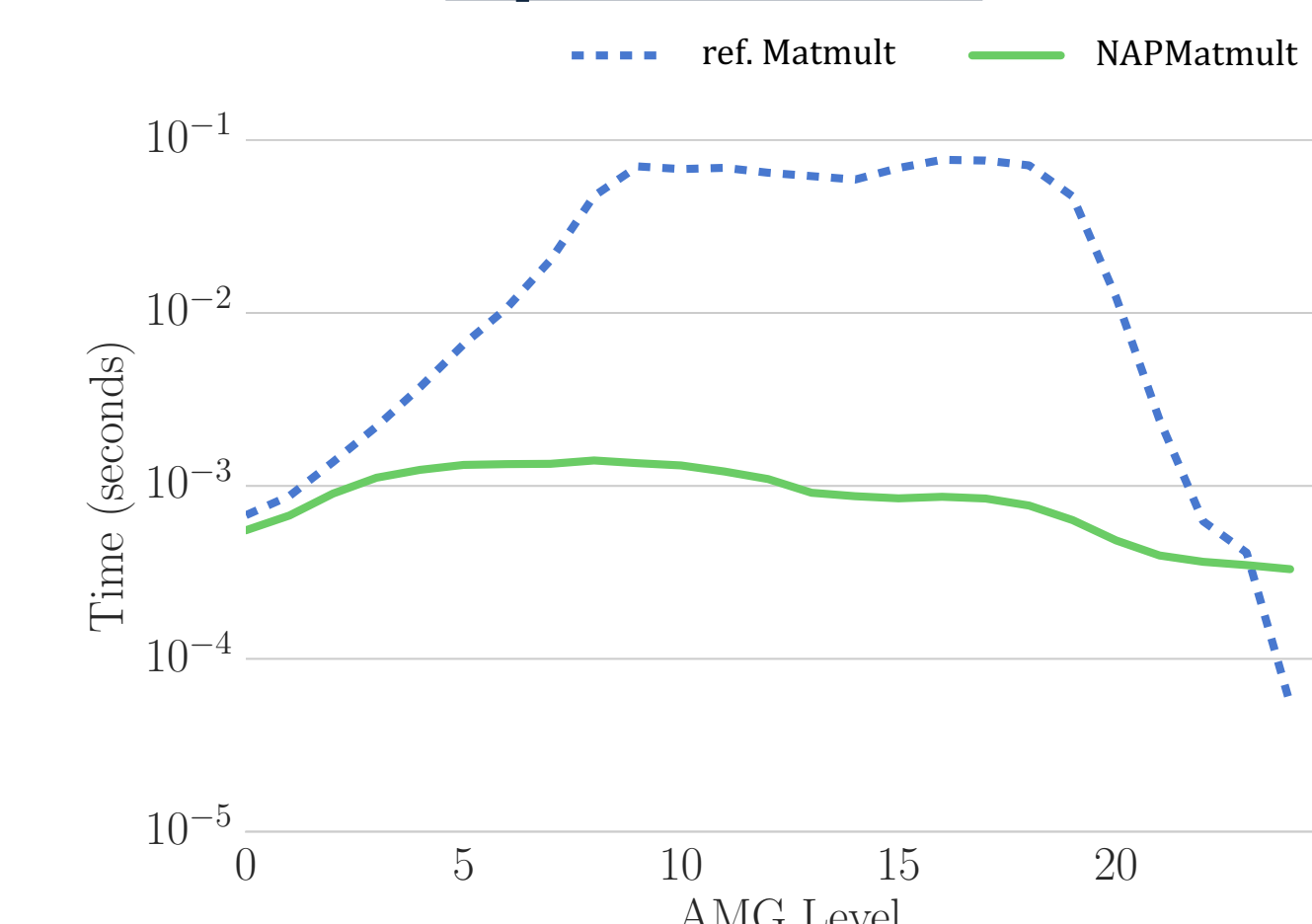


Reduces number inter-node messages (left), increases number of intra-node (right)

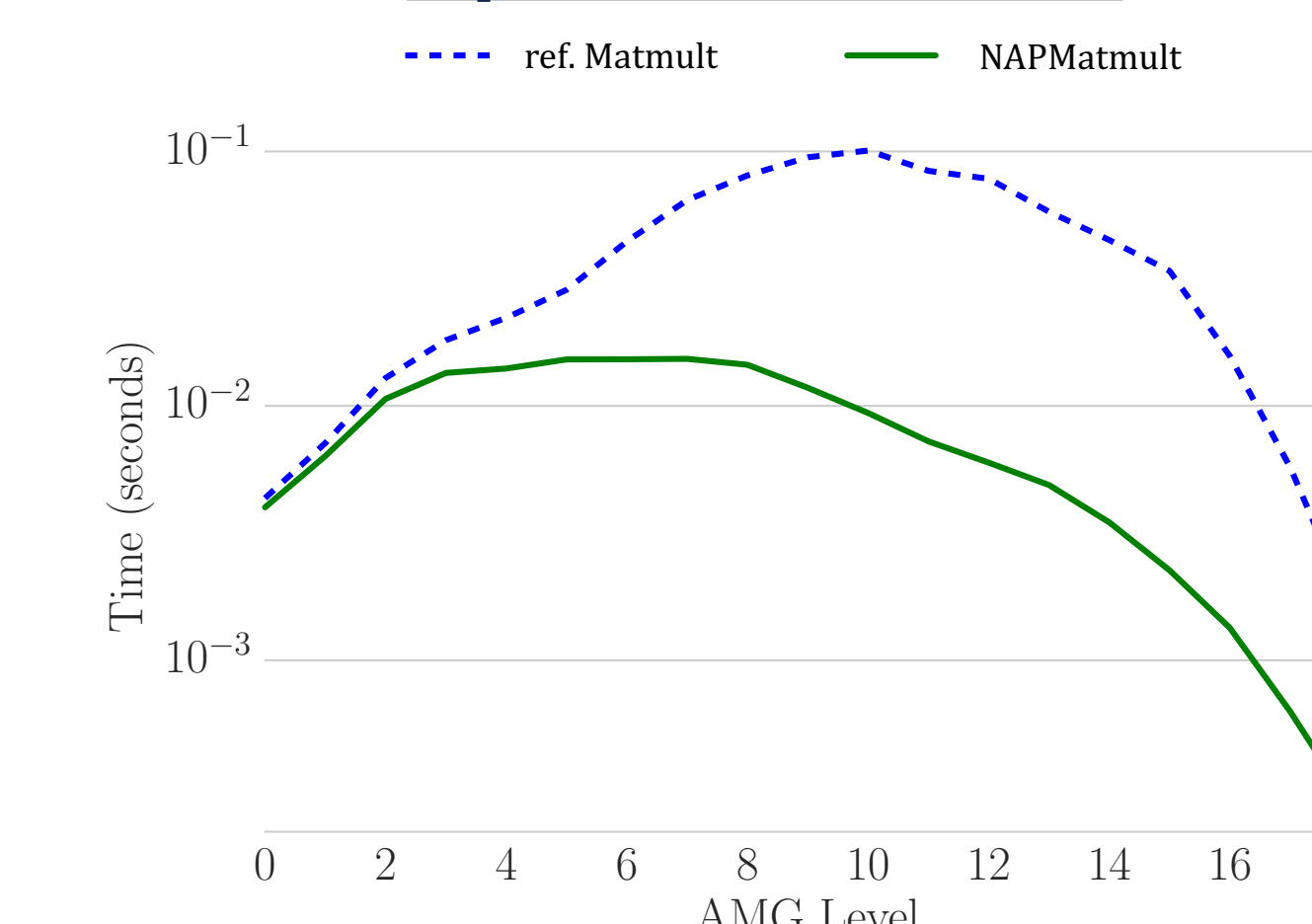


Reduces size inter-node messages (left), increases size of intra-node (right)

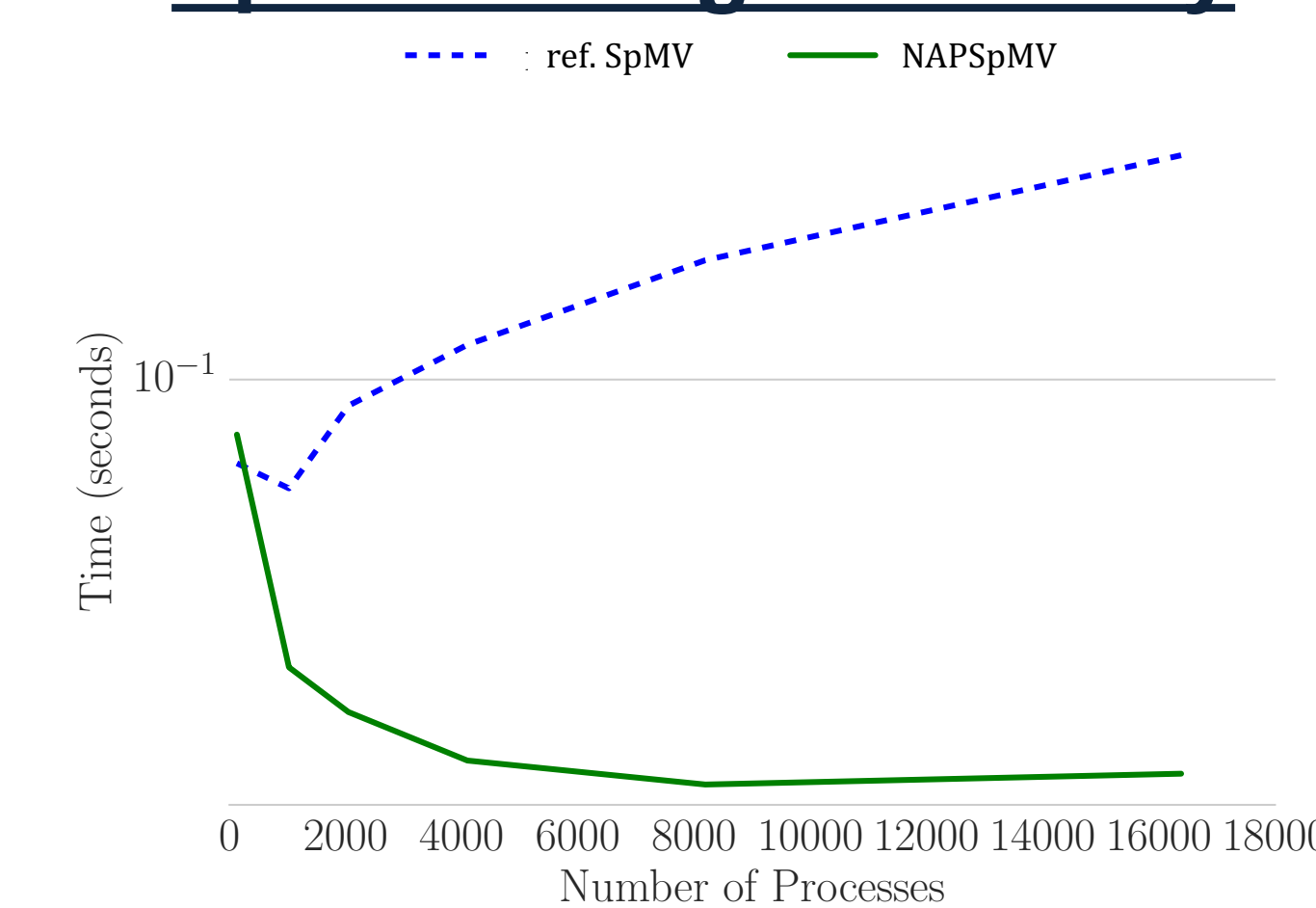
SpMVTimes



SpGEMM Times



SpMV Strong Scalability



- Greatly reduces cost of communication on coarse levels of hierarchy**
- Extends scalability (performing a SpMV on every level of hierarchy)**

Current & Future Work

- Improving performance models
- Creating parallel AMG solver, RAPtor, to be released on Github
- Analyzing topology-aware communication in AMG

Contact Information
bienz2@illinois.edu
http://web.engr.illinois.edu/~bienz2/

