Reducing Communication in Parallel Algebraic Multigrid

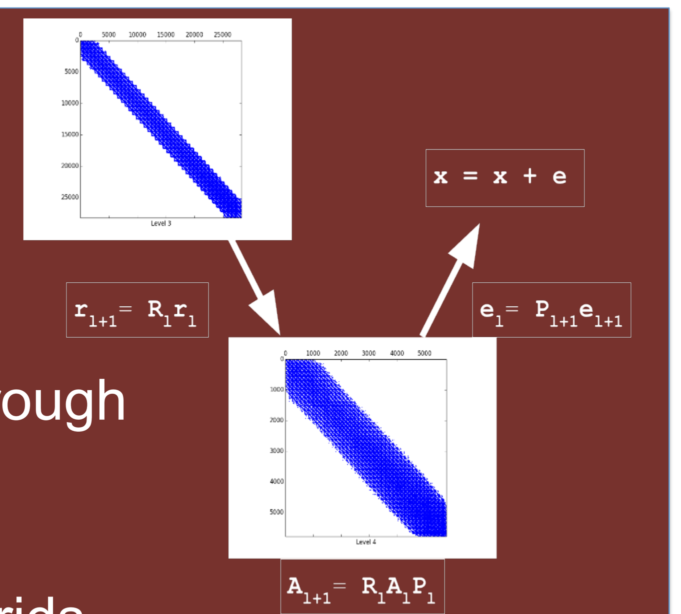
Amanda Bienz[1], Luke Olson (Advisor)[1]
[1]University of Illinois at Urbana-Champaign

Abstract

Parallel algebraic multigrid (AMG) suffers from increasingly dense communication patterns, and hence network contention, due to coarse-grid fill. This project investigates methods of improving overall performance of AMG through the tradeoff of convergence rate and communication costs.

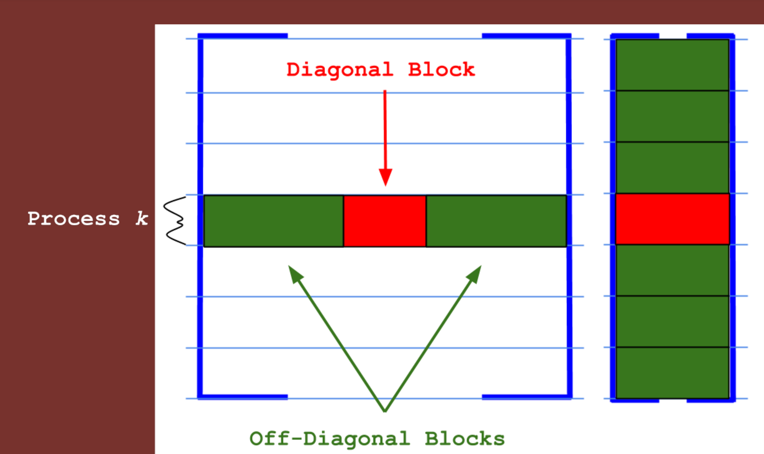
Algebraic Multigrid (AMG)

- Optimal linear solver for sparse systems
- Composed of sparse matrix-vector multiplication
  - Relaxation dampens high-energy error
  - Remaining low-energy error reduced through coarse-grid correction
- Galerkin coarse-grid product
  - Fill-in, amplified on successive coarse grids



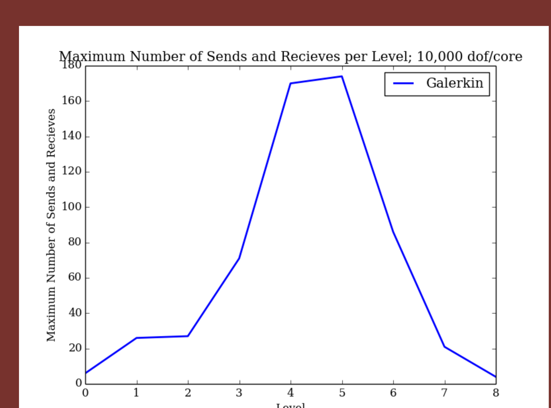
Motivation: Parallel Communication Costs

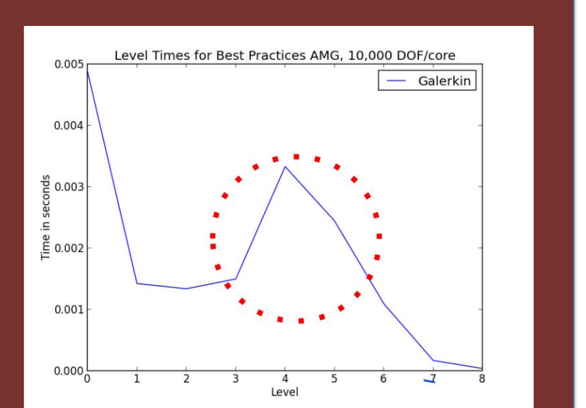
- Contiguous section of rows stored locally
- Coarse-grid fill
  → off-diagonal entries
  → communication



Problem:

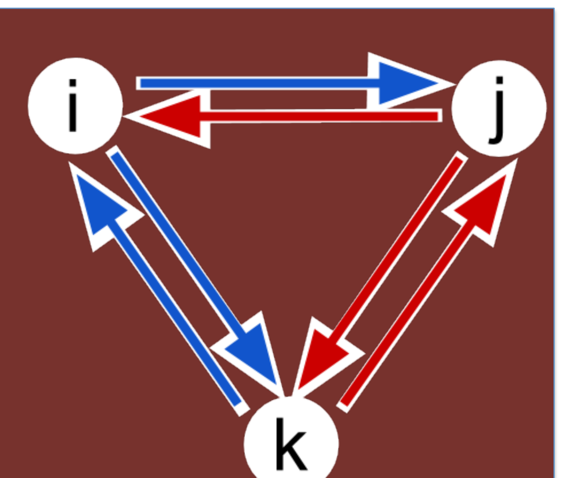
- Communication increases with problem size
- This can lead to network contention





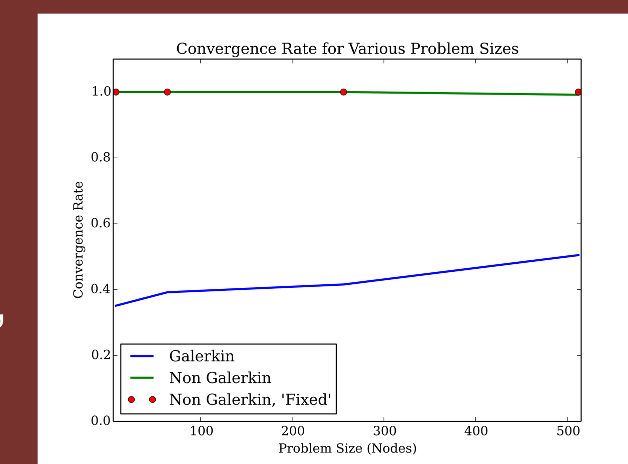
Non-Galerkin Coarse Grids[1]

- **Goal:** Reduce network contention with:
  1. Fewer number of messages
  2. Smaller average message size
- **Algorithm:**
  1. Form Galerkin coarse-grid
  2. Maintain minimal sparsity pattern for convergence
  3. Remove all entries with magnitude below tolerance
  4. Collapse removed value to strong connections



Motivation: Safer, Slower Alternative

- Optimal drop tolerance varies with problem size
- Effect on convergence is unknown in setup phase
- In solve phase, if converging too slowly, entries can be added back
- Cannot guarantee a convergence fix, because removing from fine level affects all other levels

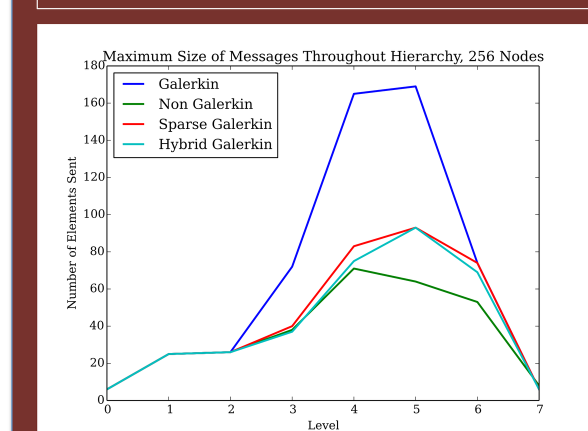


Convergence rates: Poisson 10,000 dof/core, before and after re-introducing removed entries to Non-Galerkin operators

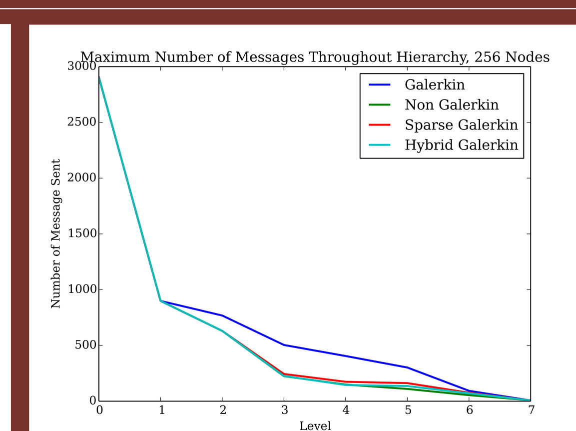
Alternative Non-Galerkin Approaches

- **Algorithm:**
  1. Form entire Galerkin hierarchy
  2. Sparsify each level with method of non-Galerkin coarse grids
- **Sparse Galerkin:** minimal sparsity pattern based on Galerkin A
- **Hybrid Galerkin:** minimal sparsity pattern based on sparsified A

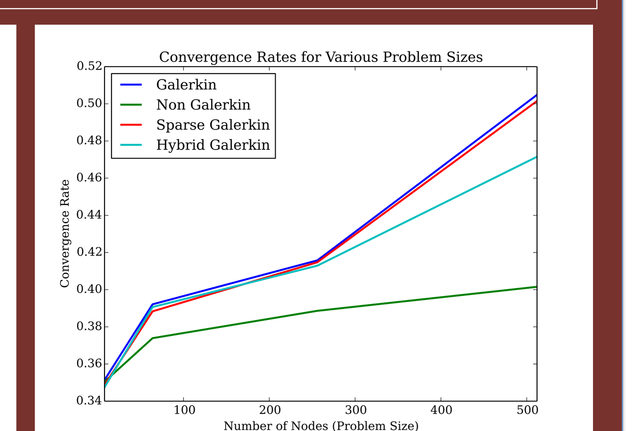
Poisson Problem: Vulcan, 8-512 Nodes



Maximum size of messages is reduced greatly, similar to original non-Galerkin method

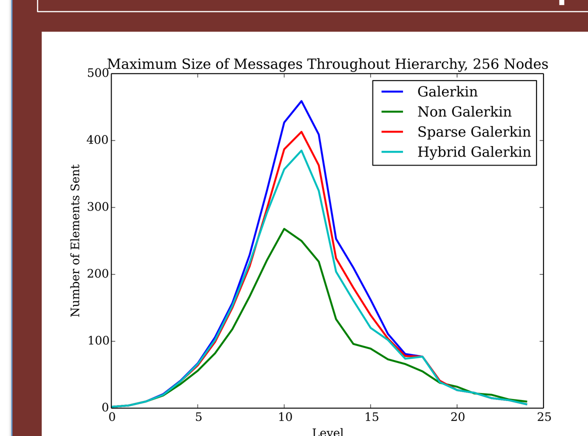


Number of messages sent is slightly reduced, similar to original non-Galerkin Method

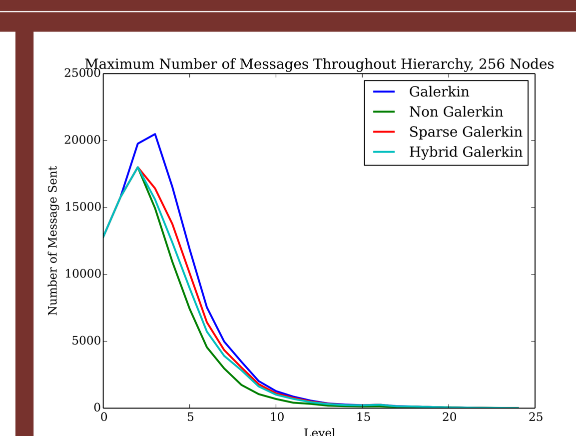


Convergence rates stay similar to Galerkin convergence rate

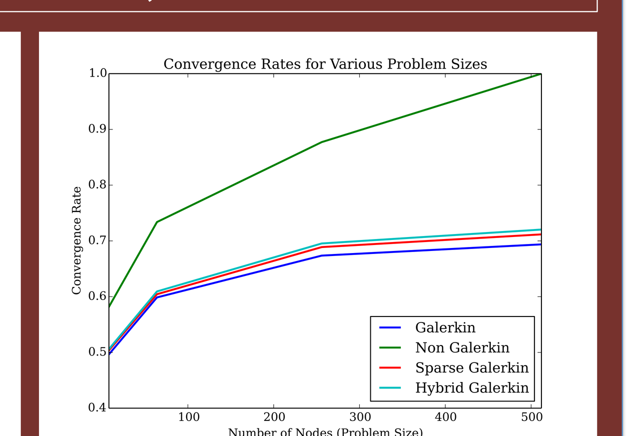
Rotated Anisotropic Diffusion Problem: Vulcan, 8-512 Nodes



Maximum size of messages is not reduced as much as non-Galerkin



Number of messages sent is similar to original non-Galerkin



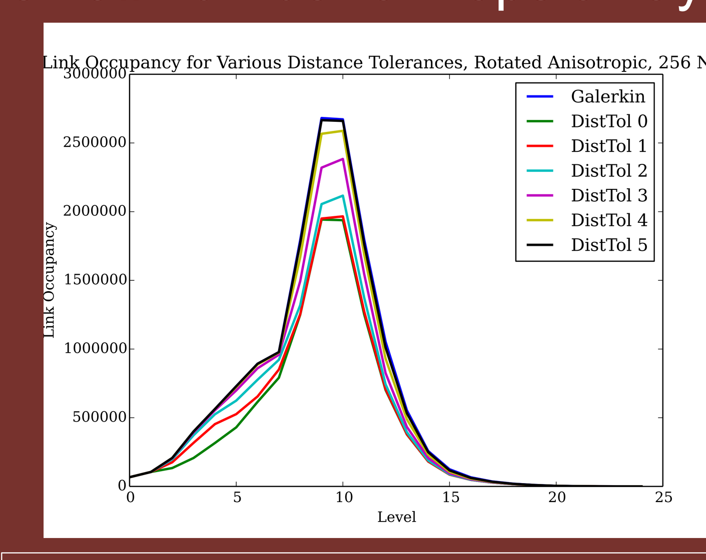
Convergence rates is much better than original non-Galerkin

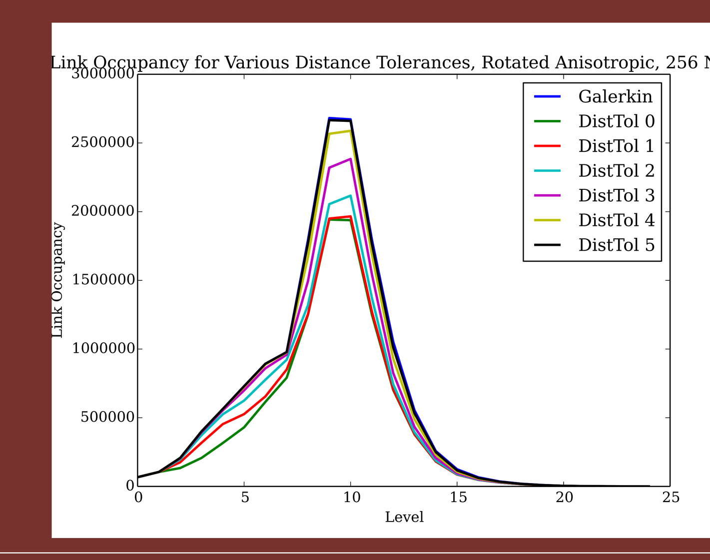
Topology-Aware Methods[2]

- **Goal:** Apply topology aware method to further reduce network contention with fewer long-distance messages
  - Convergence requirements can be relaxed for alternative non-Galerkin approaches, so long distance messages may be removed even if associated elements fall in minimal sparsity pattern
  - Remove fewer messages, targeting expensive ones
- Minimizing link occupancy – the number of packets sent, times the distance the packets travel – will reduce the network contention

Topology-Aware Results

- Topology-Aware non-Galerkin was run on Vulcan
  - Problem sizes: 8 – 512 nodes
  - 10,000 degrees-of-freedom per core
- Link occupancy, and hence network contention, can be reduced without removing communication to processors a small number of hops away





Link occupancy for various distance tolerances. Entries are removed via the non-Galerkin only if they lie on a processor 'Dist Tol' hops away. Link occupancy is measured as the product of the number of packets sent times the distance the packets must travel.

- Non-Galerkin drop tolerance can be increased for elements with high associated communication cost
- Asynchronously updating solution values, rather than dropping entries completely, allows reducing communication costs associated with more important entries
- On Vulcan, small average hop count does not imply reduced solve time, due to backup in scheduler

Future Work

- Test on Blue Waters to see correlation between link occupancy and time to solution
- Further testing with asynchronous solution updates
- Automatically chose drop tolerances, based on trade-off between communication and convergence rates
- Add entries back when convergence falls below tolerance

[1]R.D. Falgout and J.B. Schroder, **Non-Galerkin Coarse Grids for Algebraic Multigrid**, *SIAM J. Sci. Comput*. (to appear)
[2]Abhinav Bhatele, Eric Bohm, Laxmikant V. Kale, Optimizing communication for Charm++ applications by reducing network contention, Concurrency and Computation: Practice and Experience (EuroPar special issue), Vol: 23, Issue: 2, Pages: 211-222, 2011

This work was performed in collaboration with: William Gropp (University of Illinois at Urbana-Champaign); Jacob Schroder and Rob Falgout (Lawrence Livermore National Laboratory).
This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.
This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant Number DGE-1144245.