SAND2018-7174C

Parallel Distance 2 Graph Coloring for Algebraic Multigrid Preconditioners

Sandia National Laboratories
William McLendon and Mehmet Deveci
Sandia National Laboratories, New Mexico 87123

Problem

- Algebraic Multigrid (AMG) methods are used extensively as preconditioners and solvers for problems in thermal fluids and electromagnetics.
- AMG operates by creating a sequence of increasingly coarse matrices to accelerate the solution of a fine resolution linear system.
- This coarsening phase in the Trilinos multigrid library (MueLu) uses a graph to encode the structure of the matrix to create node aggregates.
- Distance-2 graph coloring can be used to generate aggregates by selecting nodes sharing the same color as root nodes, allowing them to be formed in parallel as distance-2 coloring guarantees that no aggregates will overlap.

Approach

- Extend the Greedy Coloring algorithm described in [1] to support multiple models of parallelism.
- We must handle invalid colored nodes that can occur due to concurrency.

ColorDistance2(G)

```
iter = 0
while iter < max_iterations and num_uncolored > 0 do
    parallel_for chunk_i in 0 to num_chunks do
        GreedyColor_VB(G,chunk_i,chunk_size,colors)
    num_uncolored = FindConflicts(G,colors)
ResolveConflicts(G,colors)
```

- GreedyColor loops over sets of vertices to support GPUs and other specialized environments with limited memory per thread.

GreedyColor_VB(G,chunk$_i$,chunk_size,colors)

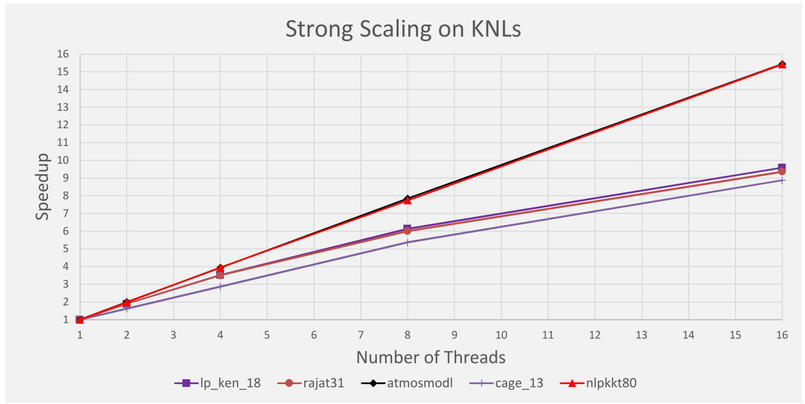
```
allocate forbidden[FORBIDDEN_SIZE]
for vid in chunk_i do
    if vid is not colored then
        offset ← 0
        while !found and offset < num_verts do
            initialize forbidden to False
            for each vid_1 in adj(vid) do
                for each vid_2 in adj(vid_1) do
                    if vid != vid_2 then
                        c = colors[vid_2]
                        if c >= offset and c-offset < FORBIDDEN_SIZE then
                            forbidden[c-offset] = True
            for each c in FORBIDDEN_SIZE do
                if forbidden[c] is False then
                    colors[vid] = offset + c
                    found = True
            offset += FORBIDDEN_SIZE
```

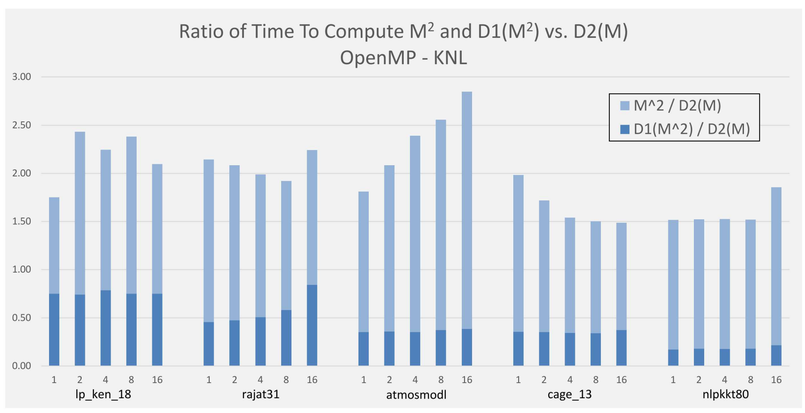
FindConflicts and ResolveConflicts are not shown due to space limitations.

Preliminary Test Results

Mesh Name	Family	Kind	Verts	Edges	Avg. Colors
lp_ken_18	LPnetlib	Linear Programming	105,127	487,008	332
rajat31	Rajat	Circuit Simulation	4,690,002	20,316,253	1252
atmosmodl	Bourchtein	Computational Fluid Dynamics	1,485,792	10,319,760	13
cage13	vanHeukelum	Directed Weighted Graph	445,315	7,479,343	118
nlpkkt80	Schenk	Optimization Problem	1,062,400	28,704,672	61

Source: University of Florida Sparse Matrix Collection





Conclusions / Future Work

- A distance-2 graph coloring algorithm developed using Kokkos Kernels for use in aggregate formation by MueLu.
- Distance-2 methods are faster than computing a distance-1 coloring of the square of the matrix.
- Future Work:
  - Explore edge-filtering techniques.
  - Test hierarchical parallelism models.
  - Develop an edge-based method?
  - Can we make it deterministic?

References & Acknowledgments

[1] What Color is Your Jacobian? Graph Coloring for Computing Derivatives. Gebremedhin, Manne, and Pothen. SIAM Review 2005 47:4, 629-705

The authors would also like to thank Nathan Ellingwood and the rest of the Kokkos team for their assistance and help, without which this work would not be possible.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under Contract DE-NA0003525.

SAND 2017-xxxP