ExaGraph: Parallel Coloring and Partitioning for Exascale Applications

SAND2019-0393PE

Erik Boman, Siva Rajamanickam, Seher Acer

ECP Annual Meeting, Houston, Jan. 2019

Date: Jan 15, 2019

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-NA-0003525.

EXASCALE COMPUTING PROJECT

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

ExaGraph: Sandia focus

Problems/motifs:

- Graph coloring
- Graph partitioning
- Branch-n-bound, combinatorial optimization (future)

Software: We will deliver through existing software

- Trilinos/Zoltan2 (distributed-memory)
- KokkosKernels (shared-memory)

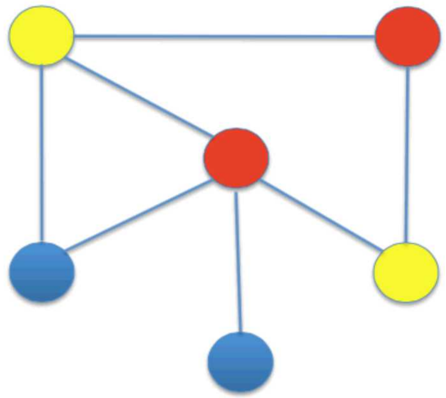
Graph coloring

- Problem:
  - Label vertices in a graph so no two neighbors have the same color.
  - Many variations
- Applications:
  - Find independent tasks for parallel computing
  - Compress sparse matrices
  - Finite differences & AD
  - Independent sets for aggregation in AMG

Graph Coloring: Problem

- Given a graph G = (V , E),
  - With vertices $v \in V$
  - Edges $(v_1, v_2) \in E \quad v_1, v_2 \in V$

Image courtesy of Sariyuce, Saule, Catalyurek, SIAM PP, 2012



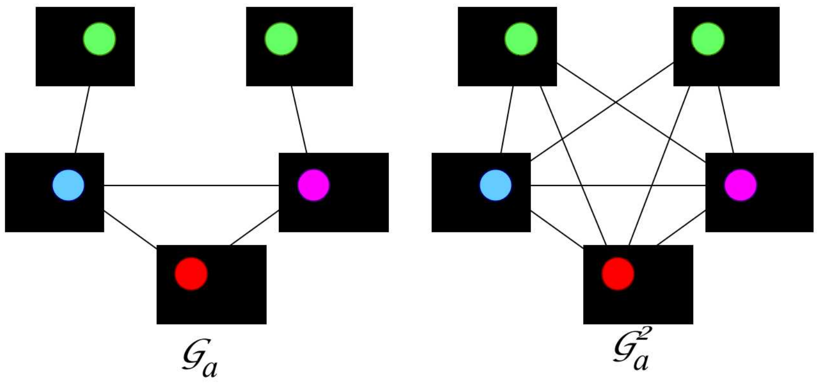
  - Distance-1 graph coloring: assign colors to vertices so that each vertex have different color from all of its neighbors $C : V \rightarrow N \quad C(v_1) \neq C(v_2) \quad for\ all\ (v_1, v_2) \in E$
- The distinct number of colors assigned to vertices: $|C|$
  - Minimize $|C|$ is NP-Hard, not practical
  - Fast greedy heuristics work well in practice
- Trade-offs:
  - Speed vs quality
  - Deterministic or non-deterministic (parallel)

Coloring and Structural Orthogonality

structurally orthogonal partition | distance-2 coloring | distance-1 coloring

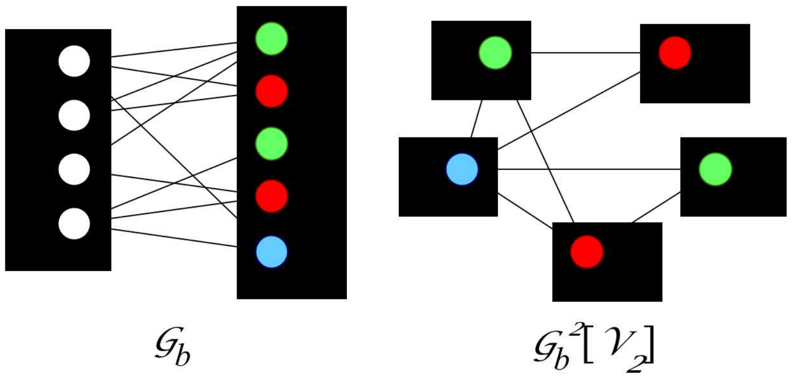
symmetric case

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 & a_{15} \\ 0 & a_{22} & a_{23} & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & a_{35} \\ 0 & 0 & a_{43} & a_{44} & a_{45} \\ a_{51} & 0 & a_{53} & a_{54} & a_{55} \end{bmatrix}$$



nonsymmetric case

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & a_{15} \\ a_{21} & a_{22} & 0 & 0 & 0 \\ a_{31} & 0 & 0 & a_{34} & 0 \\ 0 & 0 & a_{43} & a_{44} & a_{45} \end{bmatrix}$$

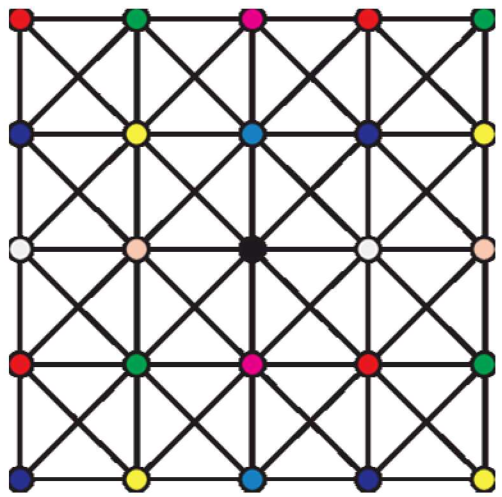


Graph Coloring Software

- ColPack (Pothen et al., Purdue)
  - Serial code (some OpenMP)
  - Wide variety of coloring problems
- Zoltan
  - MPI parallel
  - Dist-1, dist-2, partial dist-2
- KokkosKernels
  - On-node parallel using Kokkos (CPU, GPU)
  - Dist-1 and dist-2 (in progress)
- Zoltan2
  - Currently only serial coloring
  - Plan interfaces to KokkosKernels and Zoltan

Application: Aggregation in AMG

- Multigrid solvers need fast coarsening
  - MueLu uses aggregation
- Coloring produces independent sets
  - Gives seed vertices for aggregates
  - Neighbors assigned same aggregate
- Geometric multigrid:
  - Dist-1 on mesh: small aggregates
  - Dist-2 on mesh: large aggregates
- Algebraic multigrid:
  - Use matrix as mesh not available
  - Use bipartite graph in nonsymmetric case

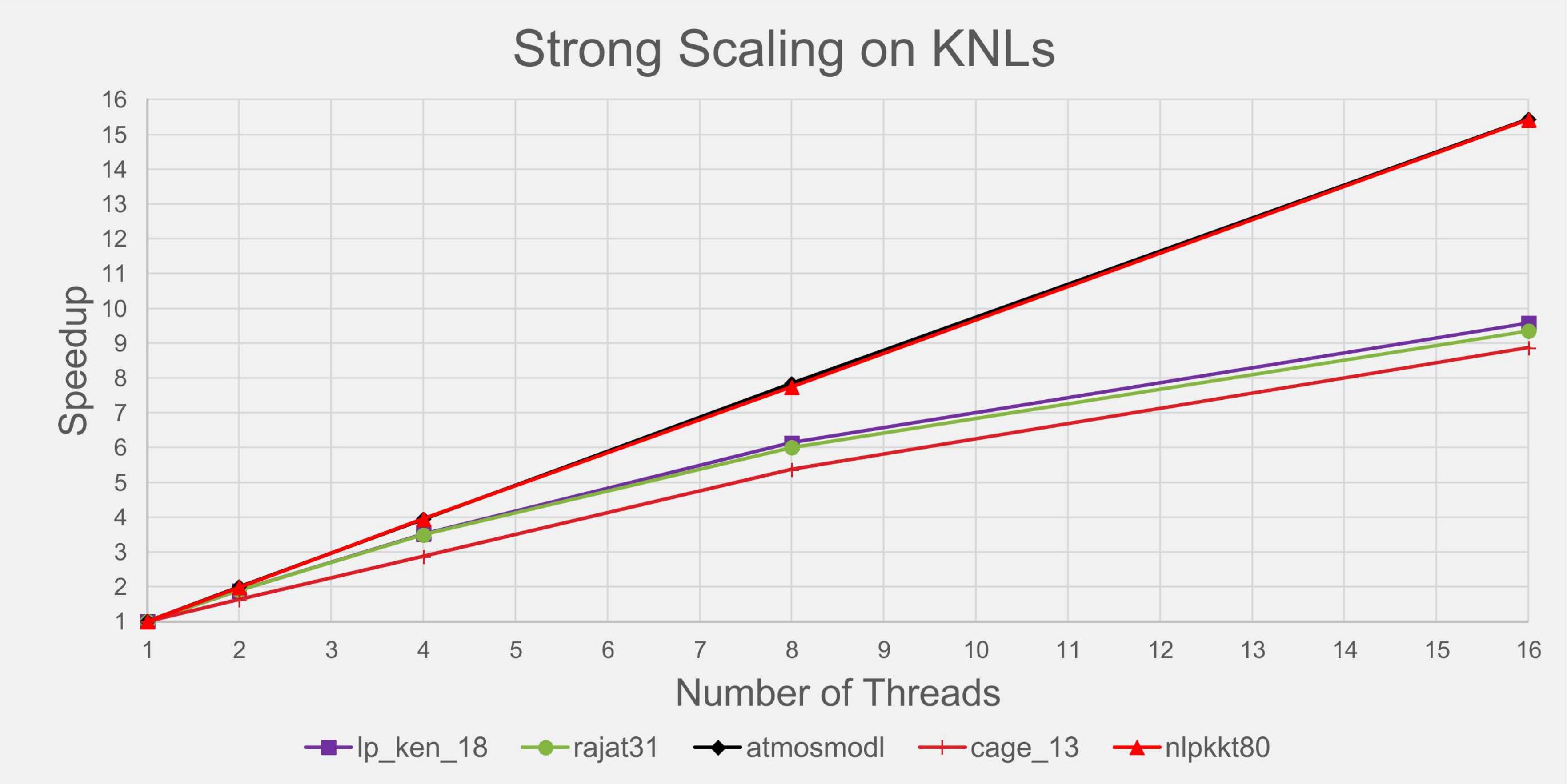


Distance-2 coloring of a mesh (credit: M. Gee)

ExaGraph Accomplishments: Coloring

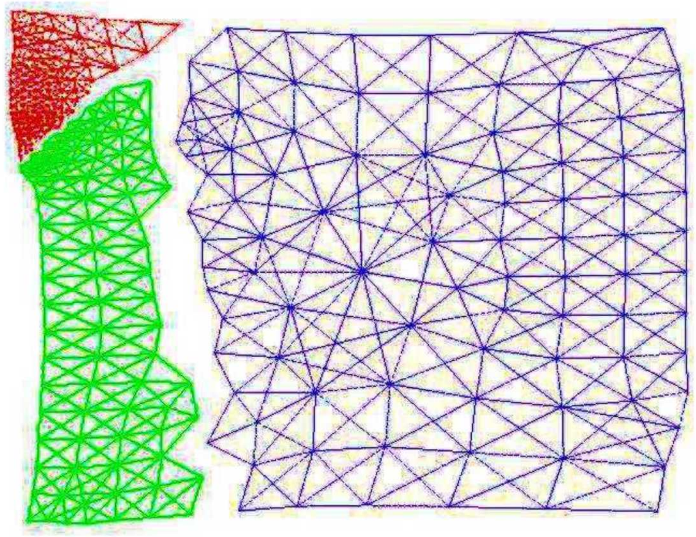
- Shared-memory distance-2 coloring using Kokkos
  - Extended previous distance-1 method (IPDPS'16)
  - Portable code runs on multicore, MIC/KNL, and GPU
- Integration with MueLu AMG solver
  - Joint work with ATDM/SNL math libraries project

Exagraph: Distance-2 coloring on KNL



Load Balancing and Partitioning

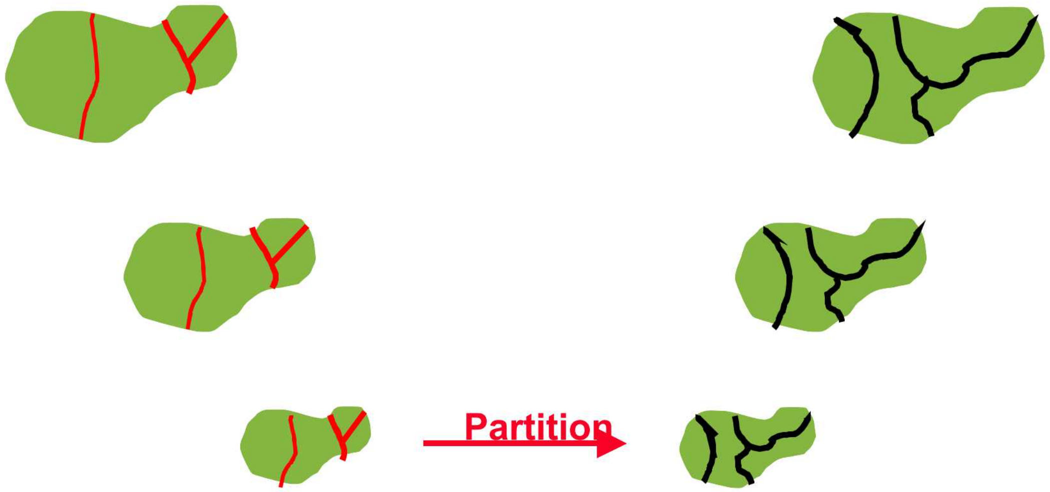
- Partitioning:
  - Assignment of application data to processors.
    - For example: mesh points, elements, matrix rows.
- Ideal partition:
  - Work (load) is well balanced among proc.
  - Inter-processor communication is kept low.
    - Low communication volume, few messages, etc.



Partition of an unstructured finite element mesh for three processors

Multi-Level Graph Partitioning

- Highly successful graph partitioning method
  - Bui & Jones (1993); Hendrickson & Leland (1993); Karypis and Kumar (1995)
  - Construct smaller approximations to graph.
  - Perform graph partitioning on coarse graph.
  - Propagate partition back, refine as needed (typically each level)
- Software:
  - Graphs: (Par-)Metis, Scotch, KaHip/Kaffpa, …
  - Hypergraphs: PaToH, hMetis, Zoltan/PHG, Mondriaan
- Parallel:
  - Algorithms are hard to parallelize
  - Crude heuristics often used
  - Quality often deteriorates for large #cores



Spectral partitioning

- Algebraic algorithm based on graph Laplacian: *min* $\frac{x^T L(G) x}{x^T x}$
  - Partitioning: Fiedler ('73), Donath, Hoffman ('73), Pothen, Simon, Liou ('90)
  - Clustering: Hagen, Kahng ('92), Shi, Malik ('00), Ng et al. ('02)
- Pros:
  - Matrix-based, can reuse linear algebra software (e.g. Trilinos)
  - Well suited to GPU and accelerators
- Cons:
  - Computationally expensive
  - Poor quality in some cases

Exagraph Partitioning (Work in progress):

- Evaluation of current partitioners
  - Perceived problem is that quality deteriorates as #cores increases
- Vertex separators for Nested Dissection
  - Most Zoltan(2) partitioners partition vertices (find edge separators)
  - Factorization based methods like SuperLU and Strumpack need vertex separators for nested dissection
  - Post-processing edge to vertex separator will work with *any* partitioner
- Scalable partitioner for multi-GPU
  - Investigating both spectral and multilevel methods

Zoltan2 Overview

- Zoltan2: Trilinos package. Toolkit of combinatorial algorithms for parallel computing on emerging architectures
- Goals:
  - Provide algorithms needed by applications on NGP
    - Load-balancing and task placement for supercomputers (hierarchical systems)
    - Node-level coloring for multi-threaded parallelism
  - Provide implementations for leadership class systems
    - Multi-threaded partitioning algorithms
  - Support very large application problem sizes
    - Templated data types for local and global indices
  - Greater integration with Trilinos' next-generation solver stack

Zoltan2 Interface

- "Adapters" describe application data to Zoltan2
  - Matrix adapter (# rows, # nonzeros, weights, row entries, ...)
    - XpetraCrsMatrixAdapter, TpetraRowMatrixAdapter
  - MultiVector adapter (# vectors, # entries, weights, entries, ...)
    - XpetraMultiVectorAdapter, BasicVectorAdapter
  - Graph adapter (# vertices, # edges, weights, adjacencies, ...)
    - XpetraCrsGraphAdapter, TpetraRowGraphAdapter
  - Mesh adapter (# entities, weights, adjacencies, coordinates, ...)
    - PamgenMeshAdapter, APFMeshAdapter
  - Identifier adapter (# IDs, identifiers, weights, ...)
    - BasicIdentifierAdapter

- Users can use a provided adapter or implement one for their application's data
  - Inherit from one of the base adapters above

Zoltan2 Capabilities: Parallel Partitioning

- Goal: Assign data/work to processors so that processor idle time and interprocessor communication are minimized
- Multi-Jagged Geometric Partitioning
  - Fast; scalable; enforces geometric locality of data
  - Uses geometric coordinates of entities
- Graph partitioning via interfaces to PT-Scotch (INRIA,France), ParMETIS (U. Minnesota), PuLP and XtraPuLP (SNL, Penn St., RPI)
  - Connectivity-based; explicitly models communication costs
  - Uses topological connections of entities
- Interfaces to Zoltan partitioning algorithms
  - Multicriteria RCB, RCB, RIB, HSFC, Hypergraph

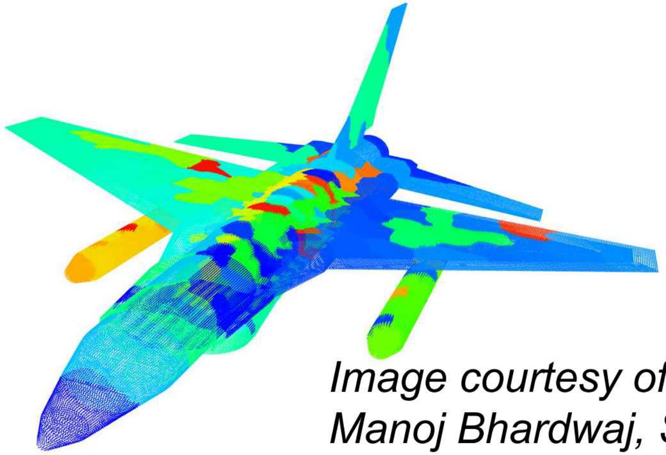
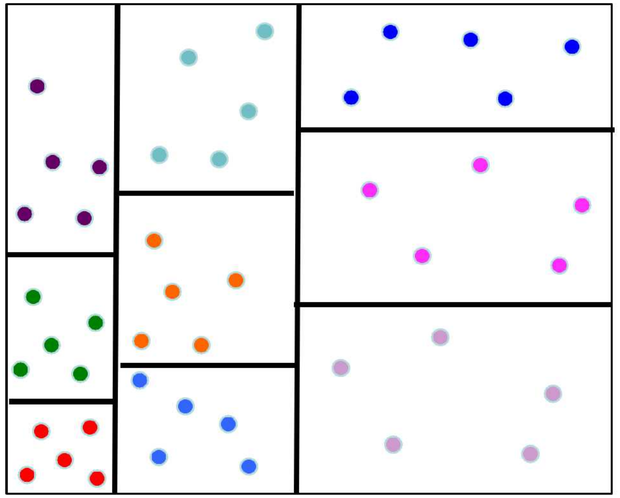


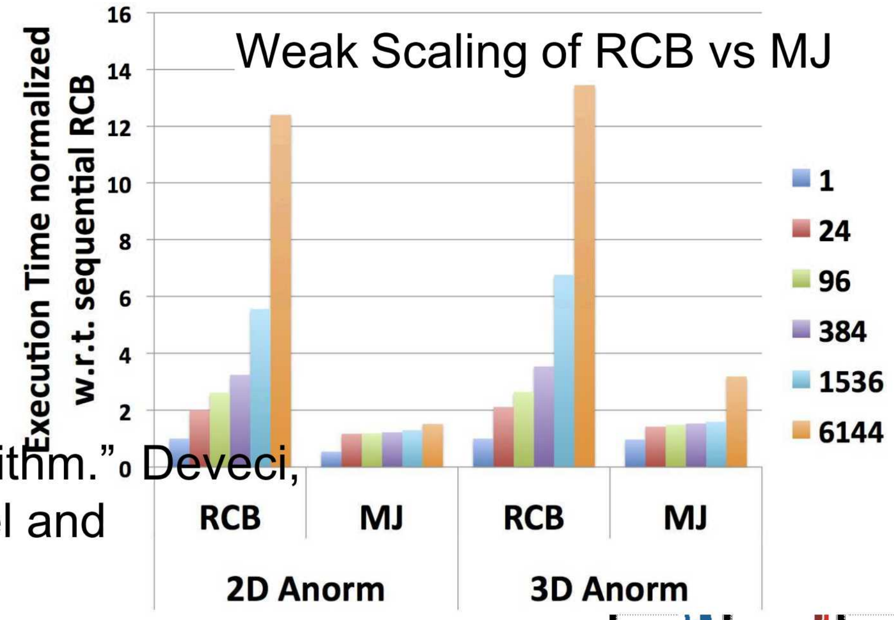
Image courtesy of Manoj Bhardwaj, SNL.

Backup

Zoltan2: Multi-Jagged Geometric Partitioning

- MPI+OpenMP implementation
  - Kokkos version in progress
- Uses multisection to divide physical domain
  - User-specified number of cuts in each dimension
- Fewer levels of recursion during partitioning
  - Less data movement than Recursive Coordinate Bisection (RCB)
  - Lower communication costs than RCB





"Multi-Jagged: A Scalable Parallel Spatial Partitioning Algorithm." Deveci, Rajamanickam, Devine, Catalyurek. IEEE Trans. On Parallel and Distributed Systems (TPDS), 2015.