Parallel Graph Algorithms in CSCAPES

Mehmet Deveci

Supervised By: Ümit V. Çatalyürek
The Ohio State University
Biomedical Informatics
Computer Science & Engineering

CSCAPES Overview

- Institute for Combinatorial Scientific Computing and Petascale Simulations which started at 2006 and funded by DOE/SciDAC
- The development, analysis, and application of combinatorial algorithms to enable scientific and engineering computations (CSC) on extreme-scale architectures

- The participants:
  - Purdue University: Alex Pothen (Lead PI)
  - Sandia National Laboratories: Erik Boman
  - Ohio State University: Ümit V. Çatalyürek
  - Argonne National Laboratory: Paul Hovland
  - Colorado State: Michelle Mills Strout
  - Totally 11 investigators, 6 postdocs and 8 students.

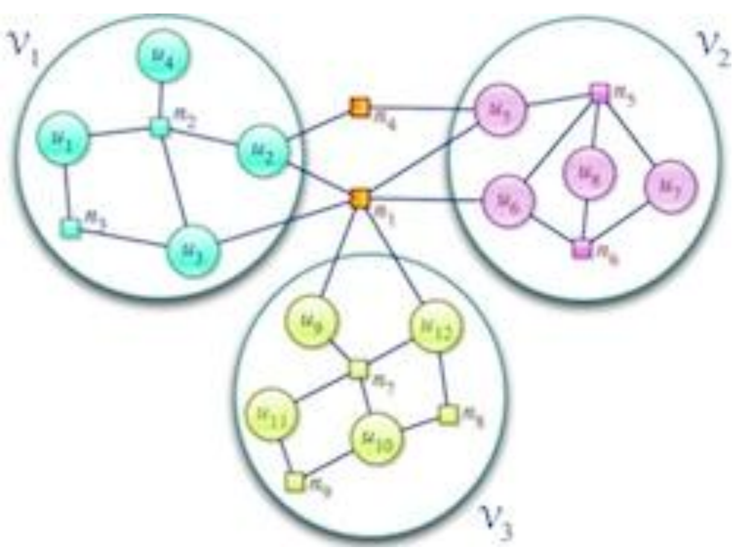
- The combinatorial problem tackled in CSCAPES:
  - Partitioning, Load Balancing, and Ordering
  - Automatic Differentiation
  - Graph Coloring in Scientific Computing
  - Matching
  - Performance Improvement via Runtime Reordering
  - Multi-scale Algorithms

  The scales: sequential, shared memory, clusters, 100K cores

Zoltan

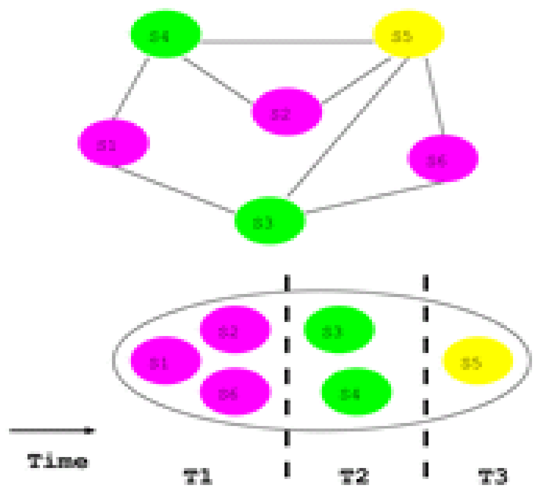
- Zoltan is one of the many software in CSCAPES projects
- A software toolkit for load balancing and parallel data management which is a C library that uses MPI but with bindings for Fortran and C++

Static and dynamic Load Balancing
Using Hypergraph repartitioning algorithms reducing communication and data migration



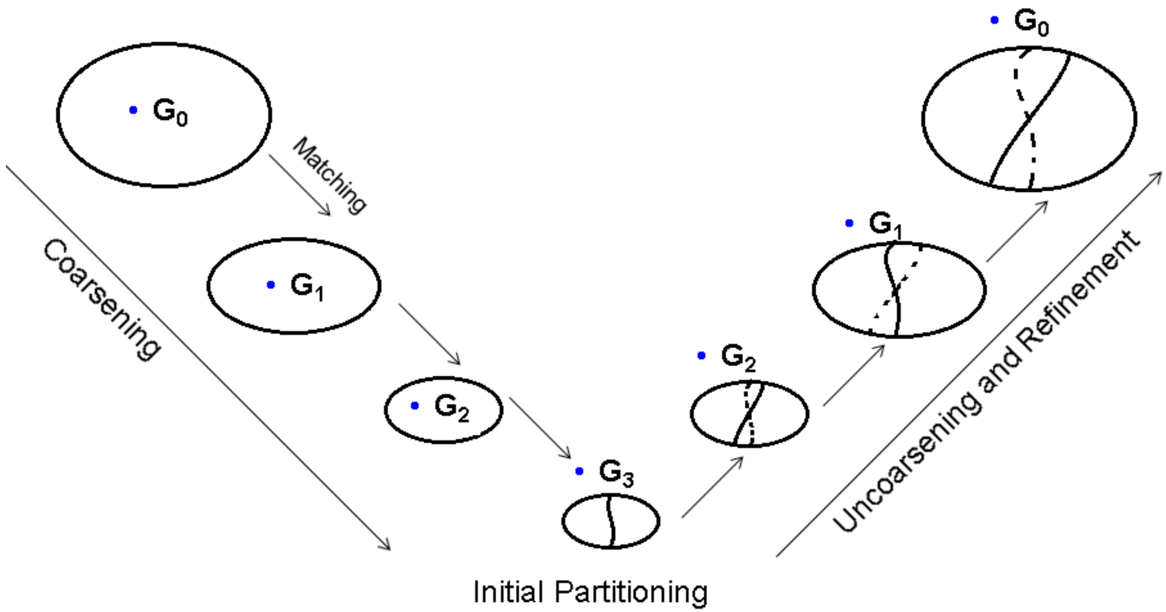
Graph Coloring
Grouping a set of objects into few independent subsets

- Sparse derivative matrix computations
- Concurrency discovery in parallel computing



Hypergraph Partitioning

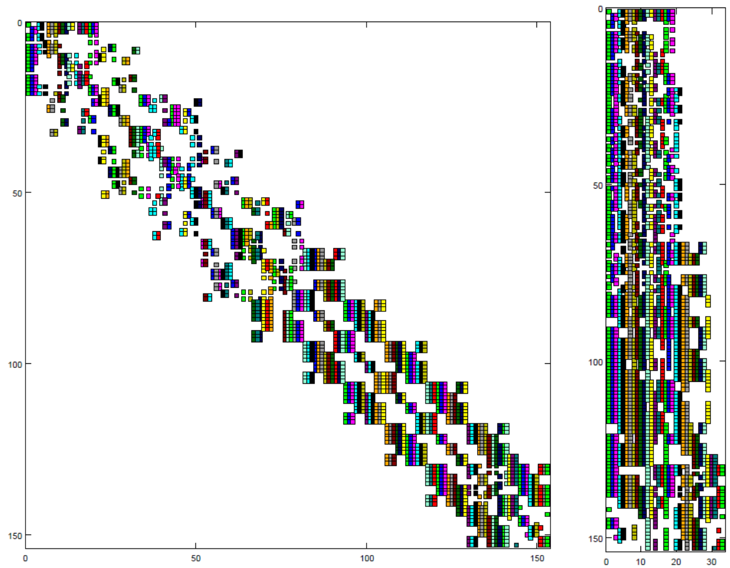
- Multilevel Hypergraph Partitioning Algorithm: V-cycle

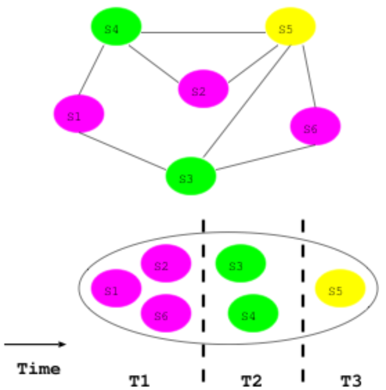


  - **Coarsening**: Iterative matching of highly connected vertices
  - **Initial Partitioning**: Partitioning of the coarsest graph
  - **Uncoarsening**: Back projection of coarsest graph to upper level, while improving the quality of partition at each iteration
- Zoltan features a distributed Hypergraph partitioner that scales to thousands of processors
- Currently implementing hybrid hypergraph partitioner to scale to 100K processors

Coloring and its applications

- Graph coloring is an abstraction for partitioning a set of binary-related objects into few "independent sets"
- Major application areas:
  - Sparse derivative matrix computation
  - Concurrency discovery in parallel computing
    - Adaptive mesh refinement
    - Iterative methods for sparse linear systems
    - Full sparse tiling



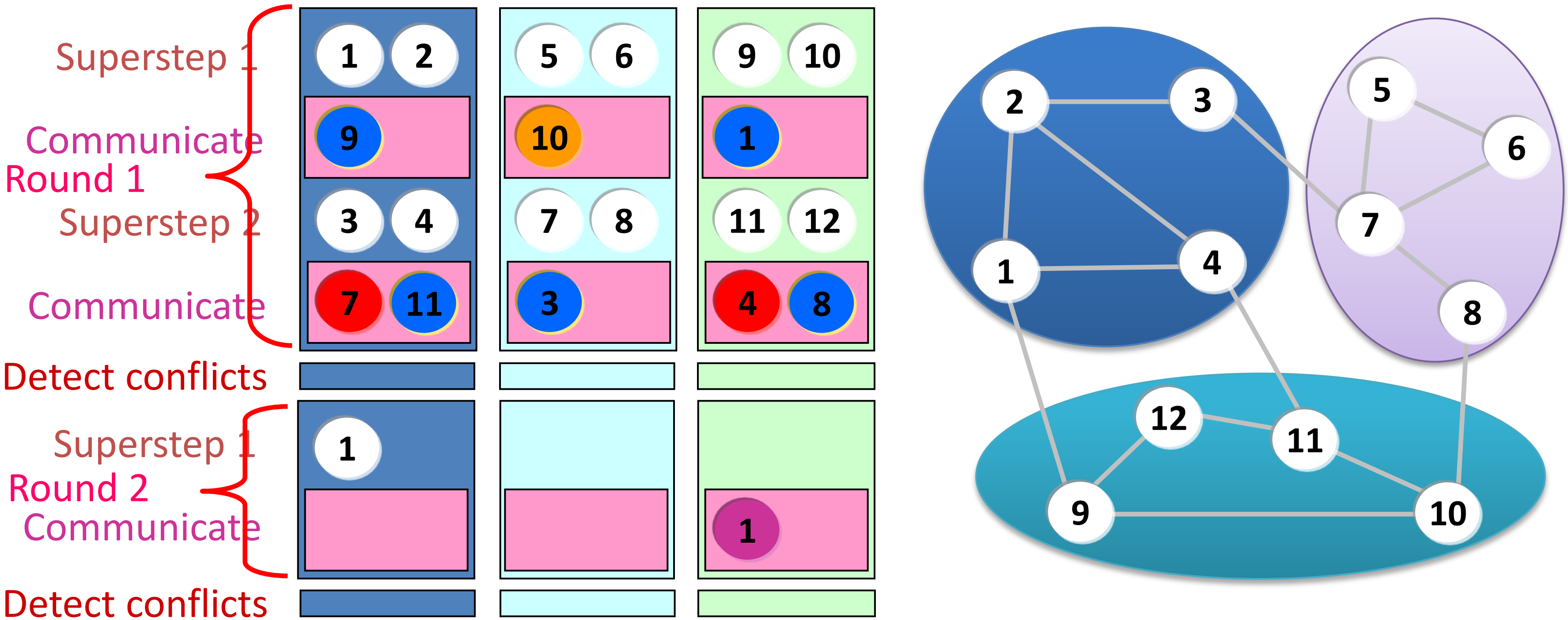


Framework for parallel greedy coloring

- Exploit features of initial data distribution
  - Distinguish between interior and boundary vertices
- Proceed in rounds, each having two phases:
  - Tentative coloring
  - Conflict detection
- Coloring phase organized in supersteps
  - A processor communicates only after coloring a subset of its assigned vertices

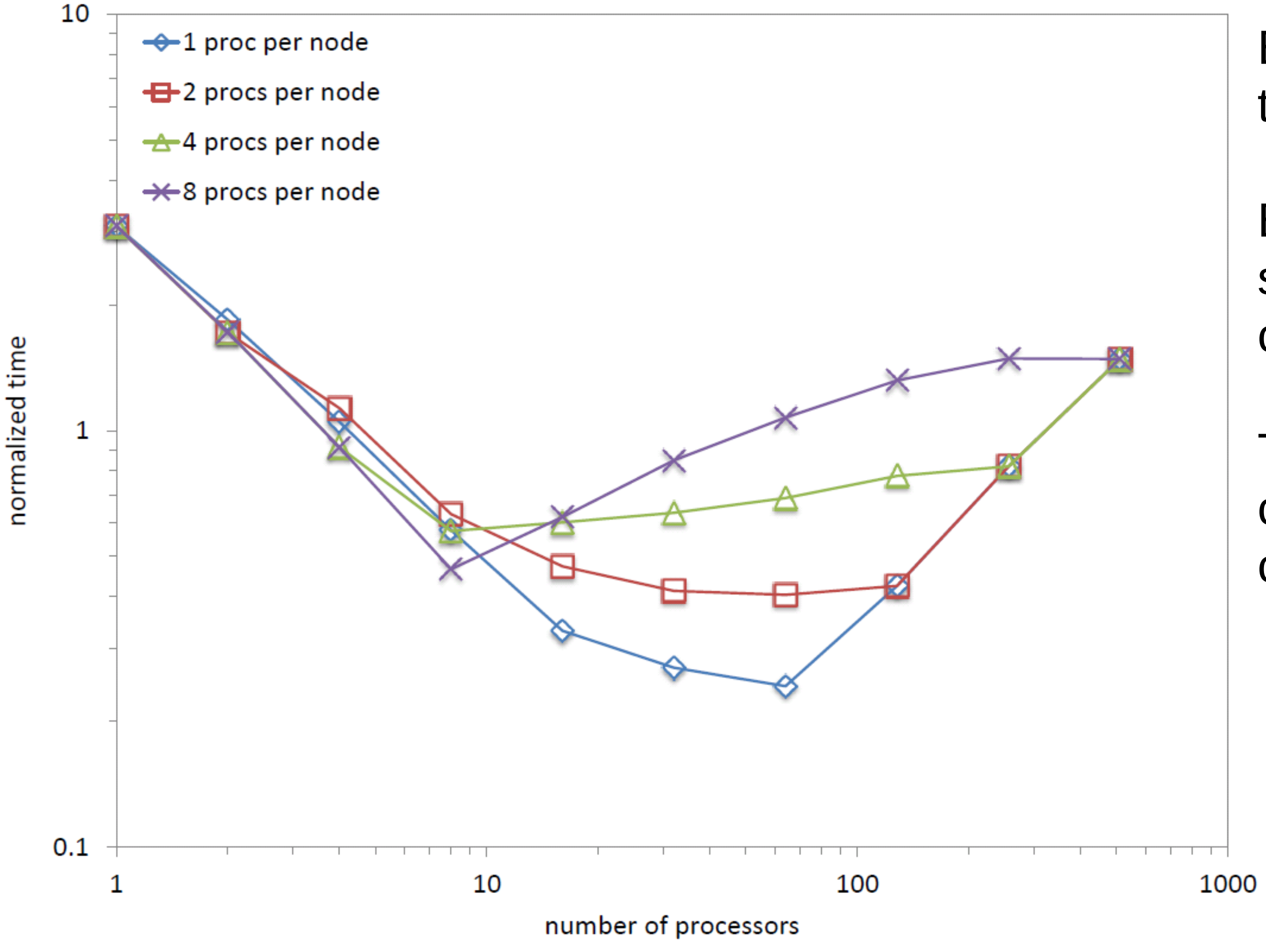
    → infrequent, coarse-grain communication

Parallel Graph Coloring



Scalability of Coloring

- Scalability of small graphs on 64 multicore nodes



100K-900K vertices and 3M-20M edges

Bad scalability when more than 1 process per node

But 8 procs per node results show that it is not a memory contention

The problem is software contention on the communication subsystem

Roadmap

- Zoltan is being rewritten as Zoltan 2 in C++ to provide better data type abstraction (e.g., 64 bit integers)
- Currently, Zoltan uses distributed programming.
- Simple partitioning and coordinate-based partitioning currently scale to 200K processors
- Moving toward hybrid programming for better scalability for Hypergraph partitioning and graph coloring

Thanks

- For more information visit
  - http://cscapes.org
  - http://bmi.osu.edu/hpc
- Research at the HPC Lab is funded by