ParCom

Parallel Algorithms for Combinatorial Scientific Computing

http://www.ii.uib.no/parcomb

Fredrik Manne
Department of Informatics
University of Bergen

Outline

- What is Combinatorial Scientific Computing (CSC)?
- Some examples
- Particular challenges when solving CSC problems on parallel computers
- The ParComb project

Combinatorial Scientific Computing

- Study of discrete algorithms in scientific and engineering applications (as opposed to continuous mathematics).
- Graph and geometric algorithms are fundamental tools.
- Has emerged as a separate field within scientific computing

Why is CSC of Interest?

- Significantly affects performance of scientific computations

  Examples

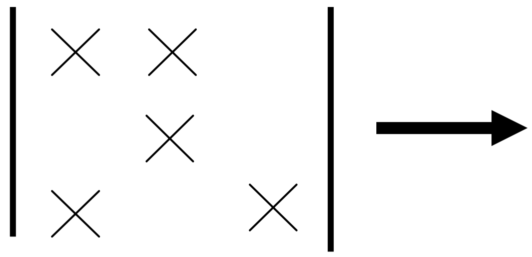
  - Increase in single processor FLOPS rate has moved problem to other areas (memory bandwidth)
  - Current increase in FLOPS rate is coming from parallel computers
  - Unstructured and sparse problems lead to complex data structures

Four Examples of CSC type problems

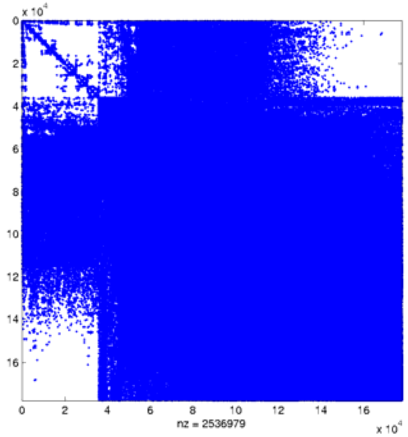
- Sparse matrix reordering
- Partitioning and load balancing
- Graph coloring in optimization
- Weighted matching

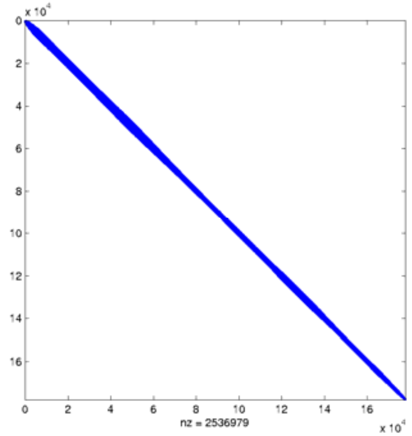
Sparse Matrix Reordering

- Reduce amount of fill elements for direct solvers



- Improve data locality

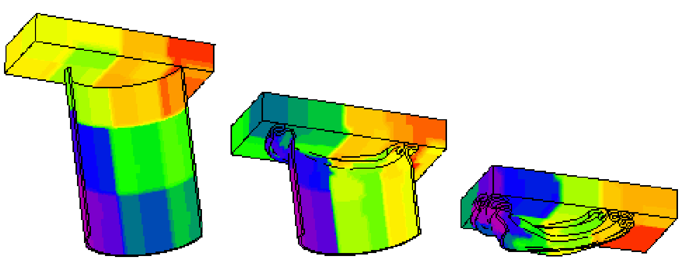




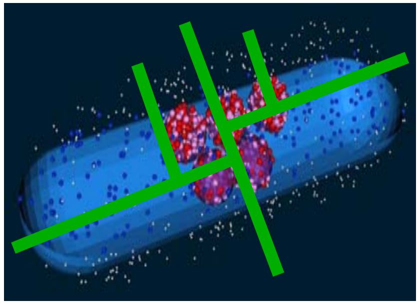
Partitioning and Load Balancing

- Goal: assign data to processors to
  - minimize application runtime
  - maximize utilization of computing resources

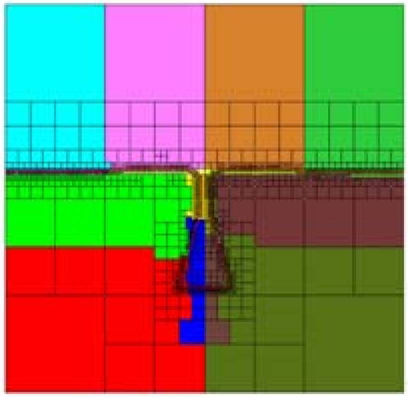
- Metrics:
  - minimize processor idle time (balance work loads)
  - keep inter-processor communication costs low



Contact detection



Particle Simulations
ParComb



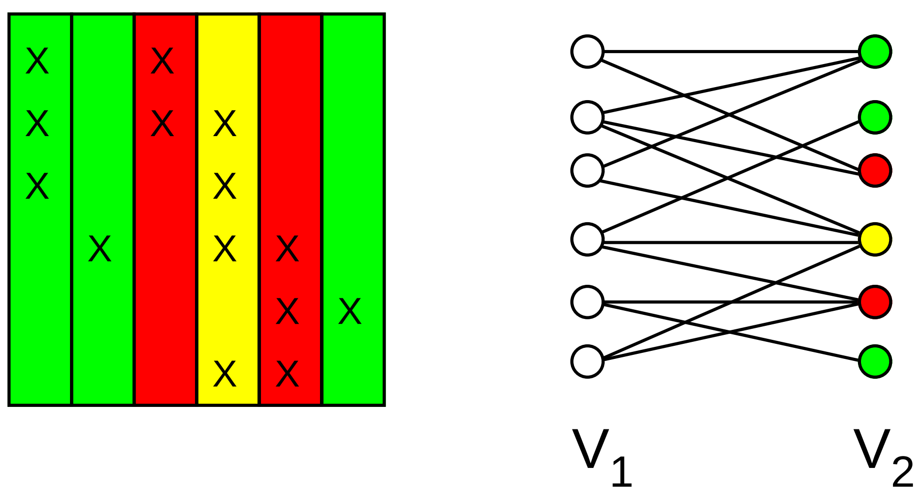
Adaptive Mesh Refinement

Graph Coloring in Optimization

Estimating sparse Jacobian matrix A using finite differences

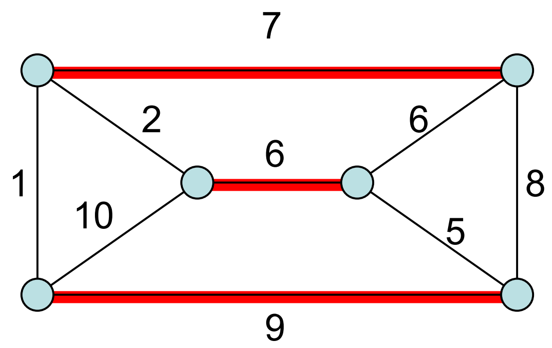
<u>Observation:</u>

-Structural independent columns can be estimated in one round of computation



Clustering columns of A $\leftrightarrow$ Distance 2 coloring of V_2

Weighted Matching



Weight = 22

Applications in CSC

- Determine pivoting strategy
- Used for clustering in multilevel algorithms

How are CSC problems (typically) solved?

- Greedy algorithms
  - Simple
  - Fast
  - Satisfactory quality
  - Inherently sequential

- On a parallel computer:

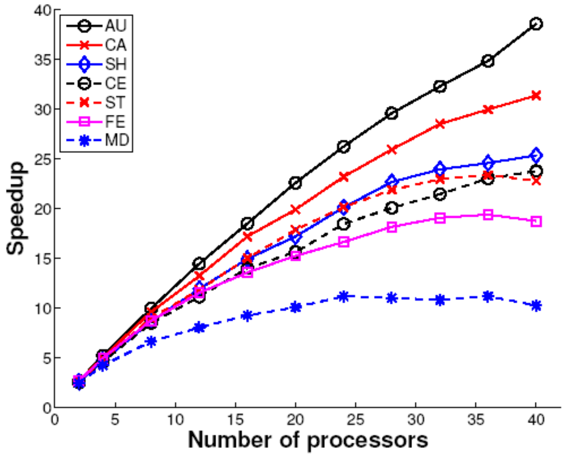
  Gather data, solve problem, spread solution

  Memory constraints and slow communication

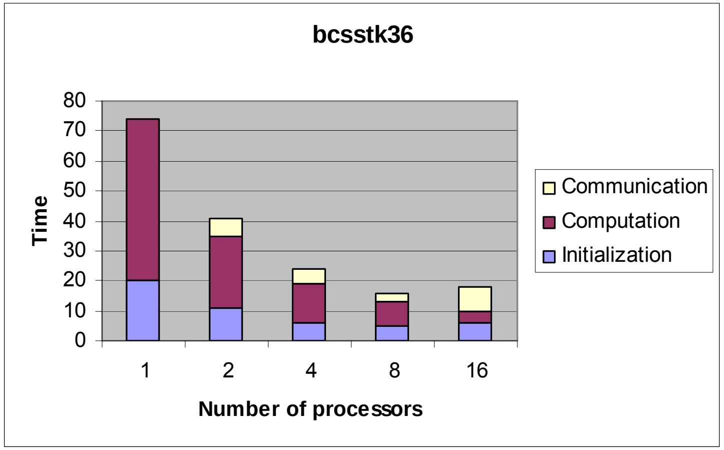
Goals

- Develop *parallel* greedy algorithms
  - Must look for independent tasks
  - Use randomization

- Some previous results:



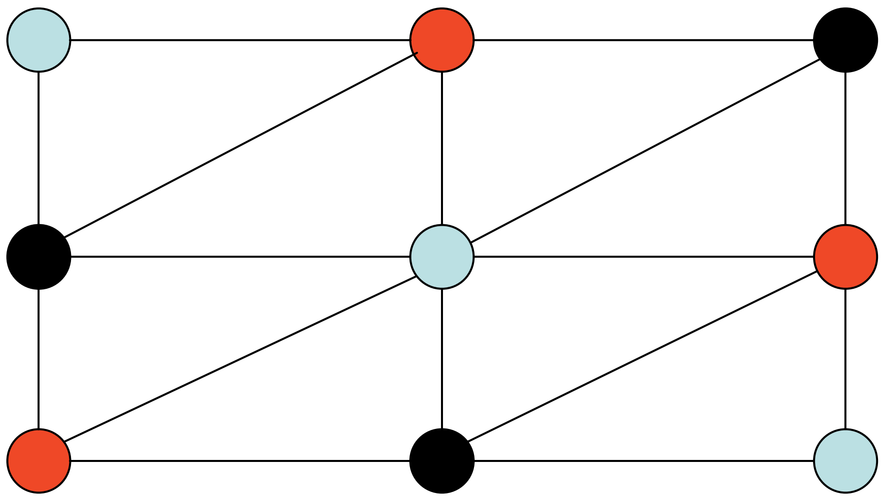
Parallel coloring algorithms



Parallel matching algorithms

A Parallel Coloring Algorithm

While G is not empty
  Do in parallel
    Find an independent set J in G
    Color and remove J from G



Objectives

- Develop new parallel methods and software for
  - Hyper graph coloring
  - Weighted matching
  - Minimal fill orderings

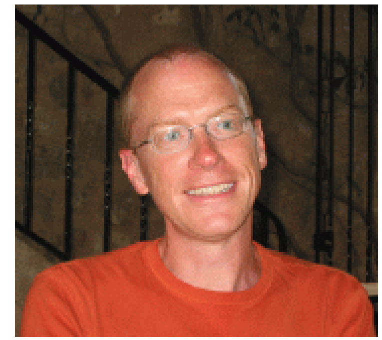
- Strong software component
- Integrate with existing packages

Participants

University of Bergen:

Petter Bjørstad
Department of Mathematics

Pinar Heggernes
Department of Informatics



Fredrik Manne
Department of Informatics

Students:

2 PhD students
-*Hossein Maserrat*
-2nd PhD fall 2007

Master students

International:

Alex Pothen
Old Dominion University

Iain Duff
Rutherford & Cerfacs

John Gilbert
Univ. of Santa Barbara

Bruce Hendrickson
Sandia Nat. Labs

Other CSC Activities

- Collaboration with the *CScapes* project (www.cscapes.org)
  - $7M project funded by DoE
  - CSC and Petascale simulations

- Comming workshops
  - SIAM Conference on Computational Science and Engineering, Los Angeles 17-19 Feb.
  - 6th International Congress on Industrial and Applied Mathematics, Zurich 16-20 July.

- More information
  - http://www.ii.uib.no/parcomb
  - http://www.cscapes.org