Spatial Sorting Algorithms for Parallel Computing in Networks

Max OrHai, Christof Teuscher
2011 October 3
Overview

- Bubble sort as a particle system
- Insertion sort in a random network

Hypothesis:
Spatial abstractions can help structure parallel computation.
Collision Sort
related work

• Cellular automata
  (e.g. Lindgren and Nordahl 1990)

• Agent-based systems
  • Particle swarm
    optimization
    (Kennedy and Eberhart 1995)
  • Ant colony optimization
    (Dorigo 1992)

• Continuous Signal Machines
  (Duchier, Durand-Lose, and Senot, SASO 2010)
Collision Sort

- Represent data as particles in a simulated continuous space
- “Bubbles” are conditional collisions
- The space may be partitioned like CA for parallel processing
Collision Sort

• Simultaneous multi-axis sorting is a natural extension

• Absolute positioning may be non-deterministic without global synchrony

• Performance depends on factors beyond particle count: speed, size of space...
Insertion Sort
(as a developmental dataflow program in an amorphous spatial computer)

related work

• Growing Point Language (Coore 1999)

• Proto (e.g. Bachrach, Beal 2006)

• Reconfigurable Asynchronous Logic Automata (Gershenfeld et al 2010)
Insertion Sort

spatial computer **assumptions** and terminology

- There are more nodes than items to be sorted
- Nodes are functionally identical
  - all run the same program
  - very limited local storage
  - no access to global information
- Nodes don’t move
- Sufficient local connectivity
- Atomic transactions
Insertion Sort

dexample sequence: extension

A.

B.

C.

D.
Insertion Sort

dexample sequence: **swelling**
Insertion Sort

amorphous network approximates a 2D manifold

**Neighbors per node at density 0.55**

- **Neighbor count**
  - 50
  - 37.5
  - 25
  - 12.5
  - 0

- **Node count**
  - 0
  - 2
  - 4
  - 6
  - 8
  - 10
  - 12
  - 14
  - 16
  - 18
  - 20
  - 22
  - 24
  - 26
  - 28
  - 30
  - 32
  - 34
  - 36
  - 38
  - 40
  - 42

- **Graphs**:
  - Radius 2
  - Green
  - Radius 4
Insertion Sort
performance and limitations

- Parallel execution yields $O(n)$ time complexity
- Growth process can get overcrowded or stuck
- No allowance for node failure in this model
- Linear linkage may be a less efficient use of space than (e.g.) spanning trees
Conclusions

• Spatial abstractions can help organize large-scale, fine-grained parallel computations

• Spatial programs may, but need not, map directly to physical computers

• Random networks can do useful work

Thanks to the Maseeh College of Engineering and Computer Science Undergraduate Research and Mentoring Program

All software models are available: http://cs.pdx.edu/~orhai