Space, Time, Power:

Evolving Concerns for Parallel Algorithms

Quentin F. Stout

Computer Science and Engineering

University of Michigan

www.eecs.umich.edu/~qstout qstout@umich.edu

February 2008

Real and Abstract Parallel Systems

- Space: where are the processors located?
- Time: how does location affect the time of algorithms?
- Power: what happens when power is a constraint?

Some Real Systems: IBM BlueGene/L

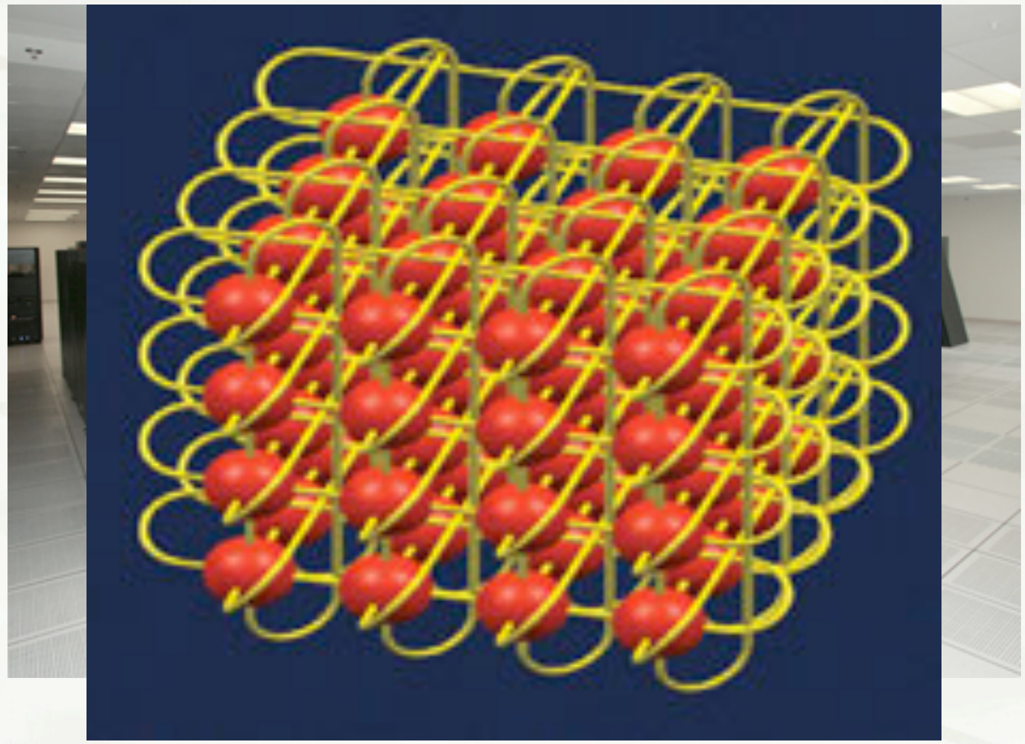
212,992 CPUs

478 Tflops

#1 supercomputer since 11/04

At Lawrence Livermore Nat'l Lab

≈ $200 Million



3-d toroidal interconnect

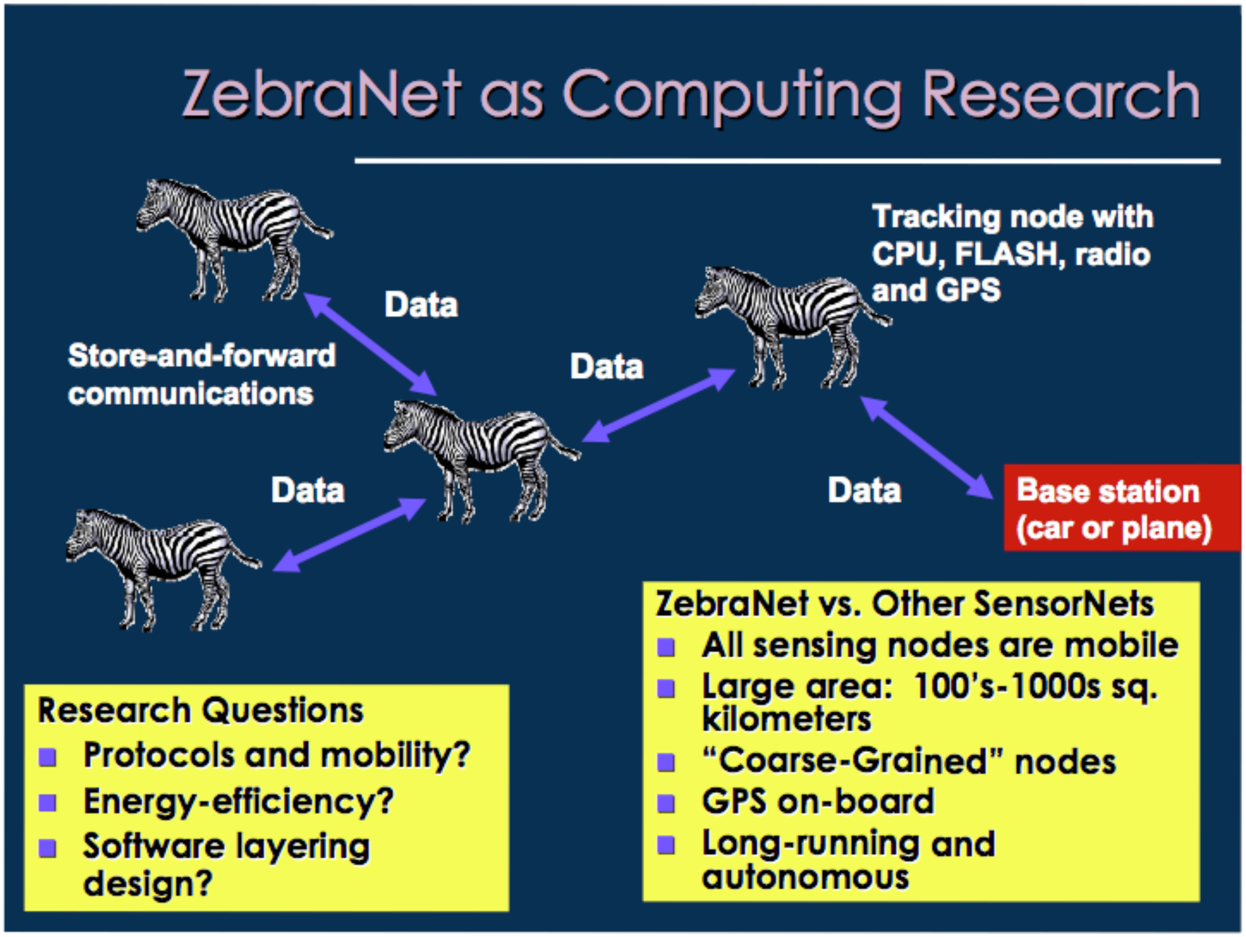
Max distance $(\#\ \text{proc})^{1/3}$

Another Real System: ZebraNet

PI

M

M
a
r
t
o
n
o
s
i



Location, Location, Location

- Processors may only be able to communicate with nearby processors
- or, time to communicate is a function of distance
- or, many processors trying to communicate to ones far away can create communication bottleneck

- Feasible, efficient programs need to take location into account

What if Space is actually Computers? Cellular Automata

- Finite automata, next state depends on current state and neighbors' states:
  location matters!

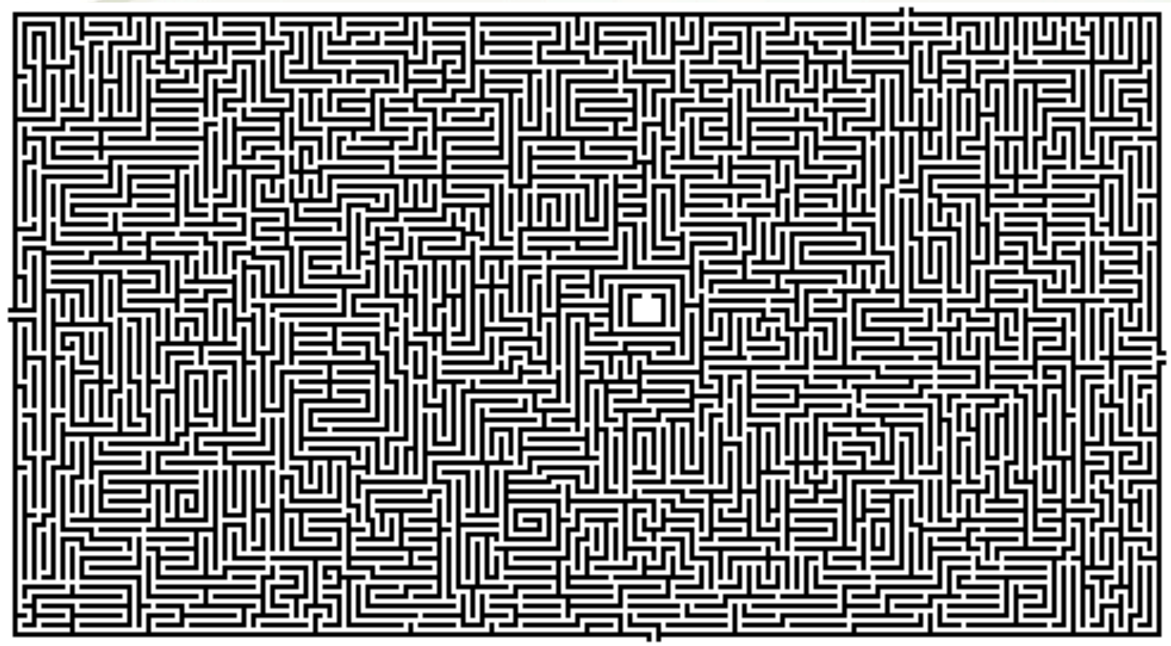
- ≈ 1950 von Neumann used as a model of parallelism and interaction in space

- Other research: Burks & al. at UM, Conway, Wolfram,…
- Can model leaf growth, traffic flow, etc.

FA	FA	FA	FA	FA	FA	FA	FA	FA	FA
FA	FA	FA	FA	FA	FA	FA	FA	FA	FA
FA	FA	FA	FA	FA	FA	FA	FA	FA	FA
FA	FA	FA	FA	FA	FA	FA	FA	FA	FA
FA	FA	FA	FA	FA	FA	FA	FA	FA	FA
FA	FA	FA	FA	FA	FA	FA	FA	FA	FA
FA	FA	FA	FA	FA	FA	FA	FA	FA	FA
FA	FA	FA	FA	FA	FA	FA	FA	FA	FA
FA	FA	FA	FA	FA	FA	FA	FA	FA	FA

Parallel Algorithms: Time

Maze of black/white pixels, one per processor in CA. Can I get out?



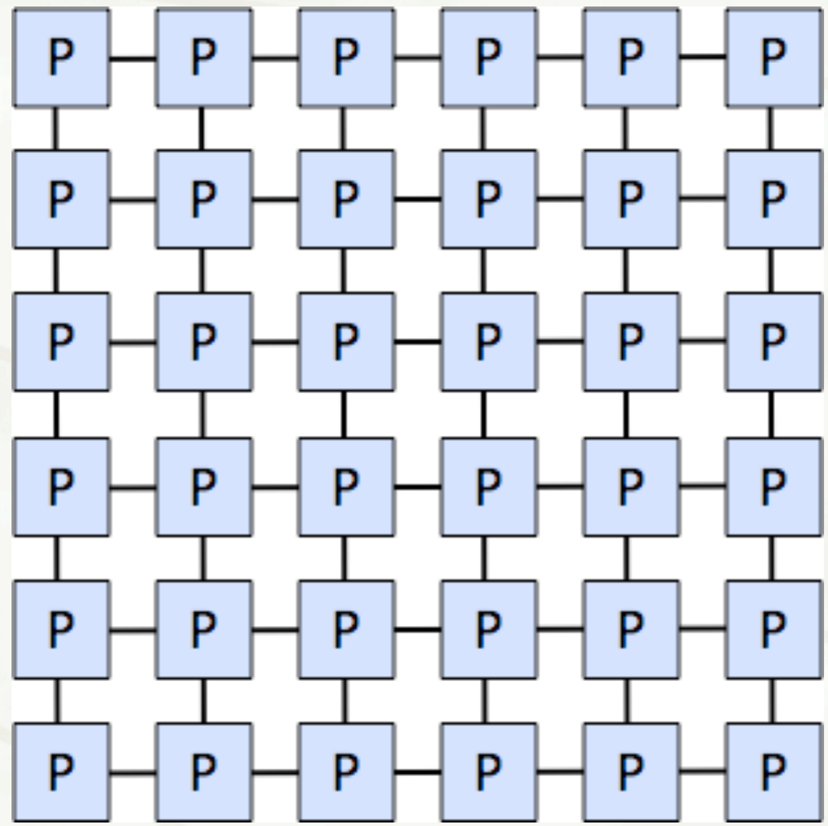
Nature-like propagation algorithm: time linear in area

Beyer, Levialdi ≈ 1970: time linear in edgelength.

CA as parallel computer, not just nature simulator

Model More Useful for Algorithms: Mesh-Connected Array

- Same model (fixed # words of memory, can exchange fixed # words at each time step)
- *But*: words $\Theta(\log n)$ bits, thus processors can store their coordinates
- *Many* algorithms known
- Theory grounded in reality: can make chips with 1000s of processors



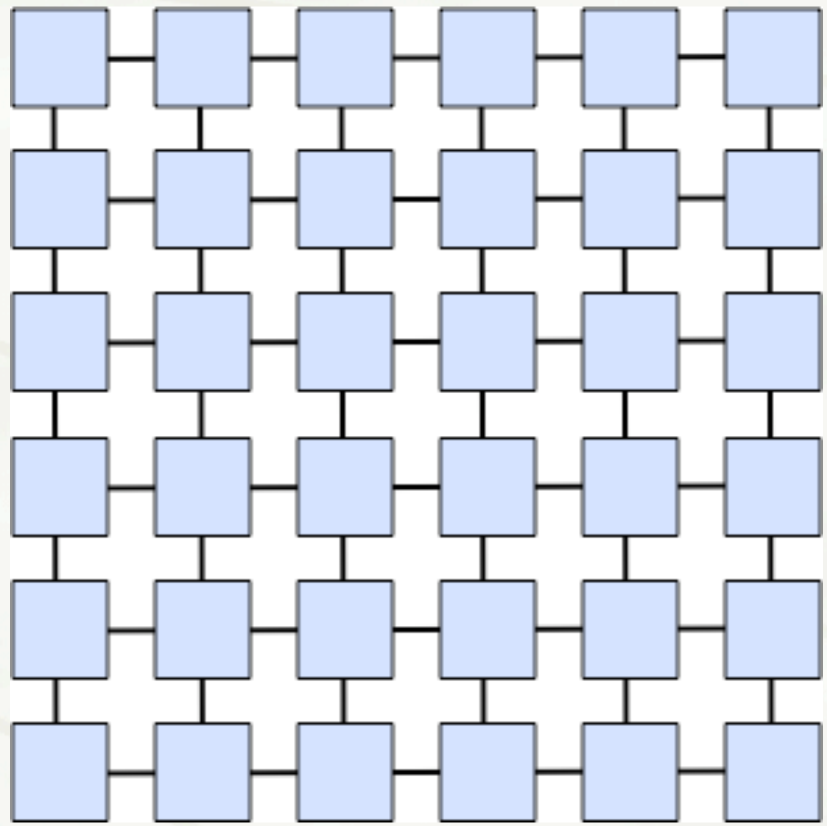
Mesh Time Bounds and Algorithms

- For square 2-d mesh with n processors, time bounds determined by spatial layout:
  - Communication radius = $\Theta(\sqrt{n})$
  - Bisection bandwidth = $\Theta(\sqrt{n})$
- Thus $\Omega(\sqrt{n})$ lower bound for any nontrivial problem. d dimensions: $\Omega(n^{1/d})$

- Sort, image labeling, matrix multiply, intersecting line segments, minimal spanning tree, etc. solvable this fast

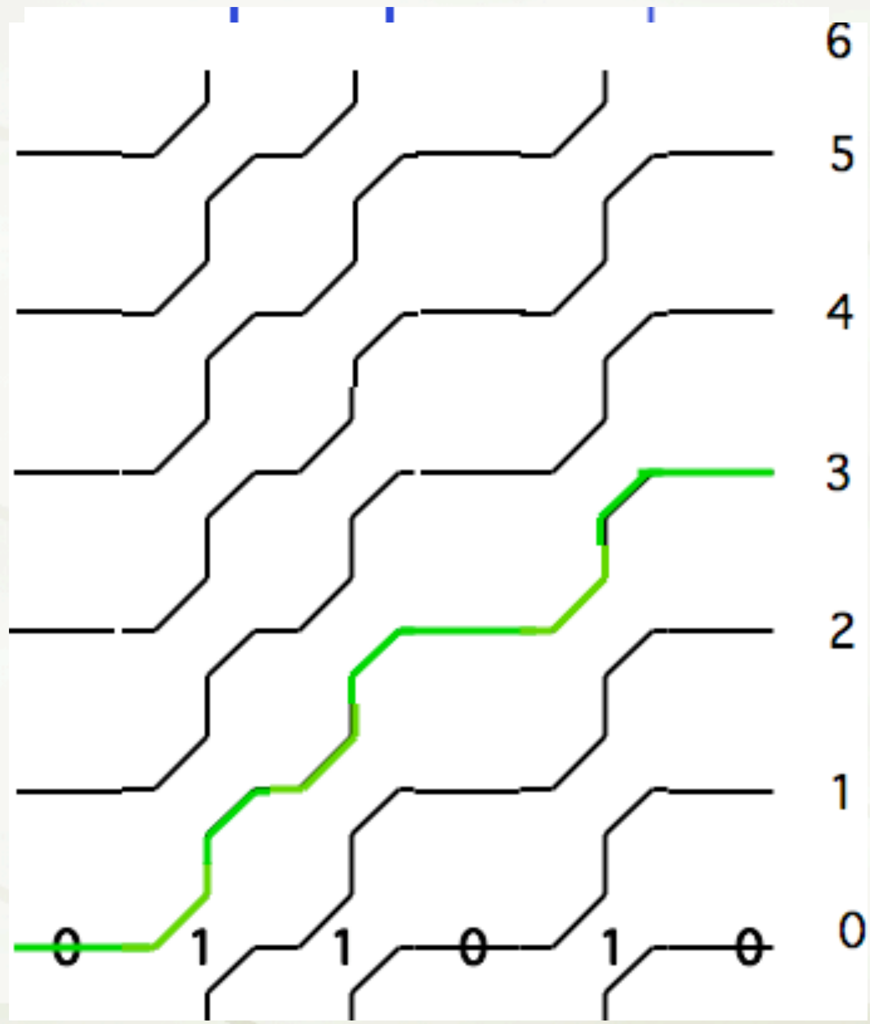
What if Space is actually Switches? Reconfigurable Mesh

- Mesh plus processors have switches to control how edges interconnect, i.e., dynamic circuits.
- Alternate rounds of compute, set switches, communicate
- Model ≈ 1985
- Have been built



Add √n Bits in Constant Time

- Bits initially in base
- Broadcast up
- 0 or 1 configuration
- Initiate signal at left
- answer at right



Reconfigurable Mesh vs. PRAM and Quantum

- PRAM
  - Parity, using poly # processors: $\Omega(\log n/\log\log n)$
  - implies summing, sorting have same bound
- Quantum
  - sort: $\theta(n \log n)$ – no faster than serial
- Reconfigurable mesh
  - parity of $\sqrt{n}$ bits $\Theta(1)$;   n bits $\Theta(\log\log n)$
  - sort $\sqrt{n}$ values: $\Theta(1)$
  - sort n values: $\Theta(\sqrt{n})$

$\Leftarrow$ bisection band unchanged, spatial layout still relevant

NEC Earth Simulator

- 5120 Processors
- 41 Tflops
- #1 supercomp
  06/02 - 06/04
- In Yokohama,
  Japan
- > $500 Million

Power
substation
≈ 20 Mw

Power Is a Concern

- Sensor networks
  - max power of any transmission
  - max power usage by any single processor
  - some work on abstract algorithms

- Supercomputers, GPUs, multicore
  - max total system power at any instant
  - abstract models, algorithms new research area

Power-Hungry Parallel Algorithms

- Previous algorithms for meshes, cellular automata, etc. assumed processors always on.
- Thus peak power = n.
- Often useful to write algorithms in terms of "data movement operations", similar to data structures.
- Unfortunately, DMOs typically involve sorting.

Power - Time Bounds

To finish in time T, must have

$$\sum_{t=1}^{T} Power(t) = \Omega(Serial\_time)$$

$\Rightarrow$ Peak Power * Parallel Time = Ω(Serial Time)

New, additional bound:

Peak Power * Parallel Time

= Ω(Total Data Movement)

Spatial location matters once again.

Mesh Power Bounds

Sorting: Total data movement (power) required: (n items) * (dist $\sqrt{n}$) = $\Omega(n^{1.5})$

$\Rightarrow$ many problems with edge inputs, point sets, etc. also $\Omega(n^{1.5})$

$\Rightarrow$ what if their input is presorted or other special arrangement?

Images, adjacency matrices: data movement and serial bounds: only bound known = $\Omega(n)$

$\Rightarrow$ can this be achieved?

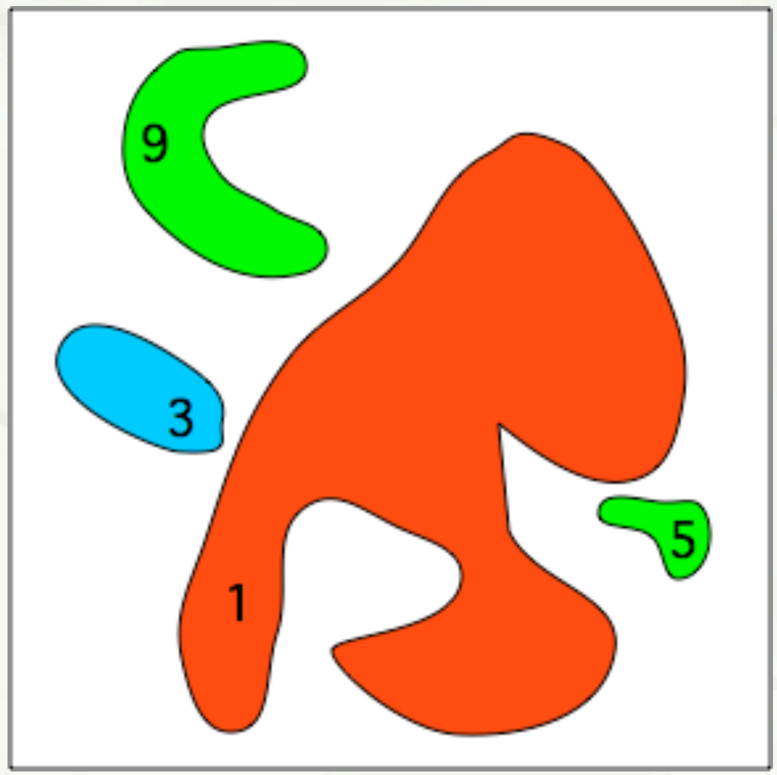
Example: Component Labeling of Image

Goal: label all pixels in each figure with a label unique to that figure

Standard parallel approach: divide and conquer

Partition image and label within each part

Reconcile local labels into global label



Power-Constrained Mesh Algorithms

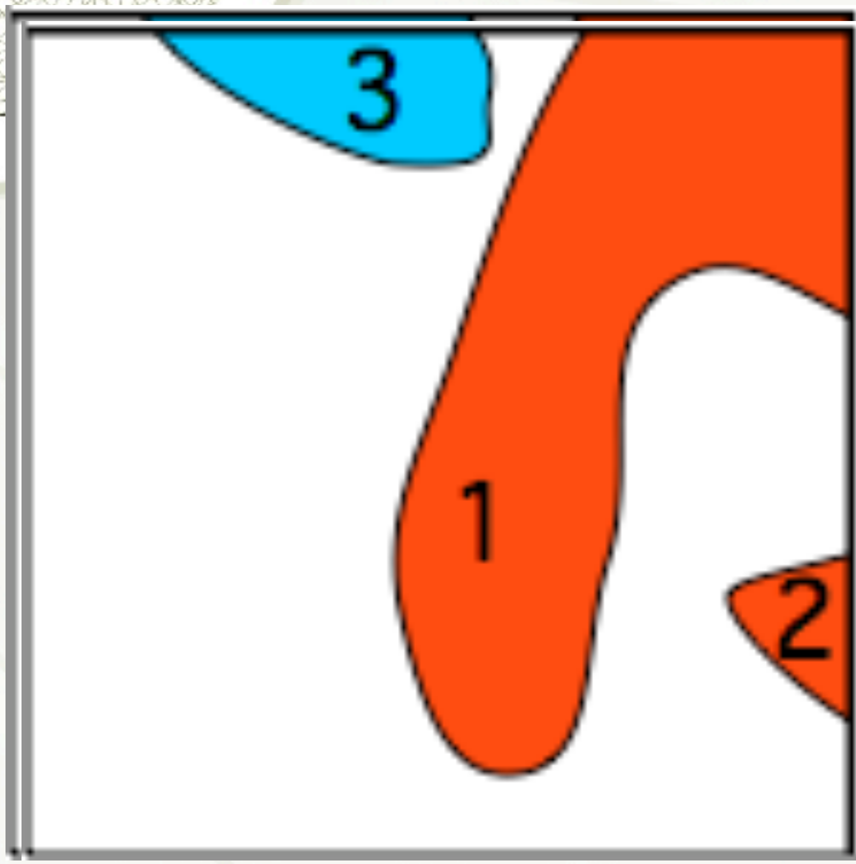
Think of rats moving around image, collecting info, storing it, cooperating to solve problem

Can carry a fixed number of words of info, can leave a fixed number at any one location.

Rat location indicates active processor, carrying info is communication

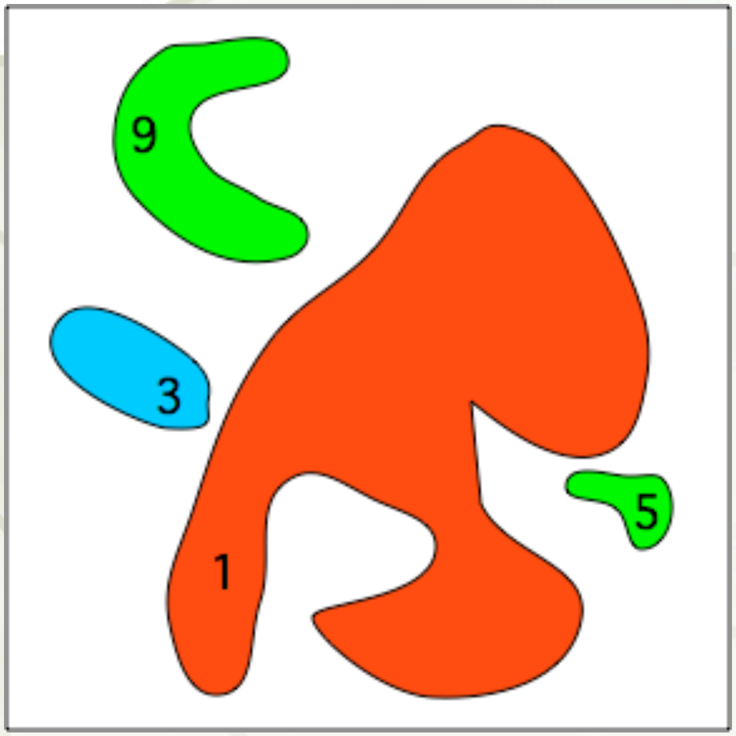
Number of rats = number of active processors at any given time, i.e., peak power

Initial Labeling



Depth-first search, linear time

Global Labeling



Bring edges to central square

Take global labels back to quadrants, update pixel labels

(2,4) (1,8)

(3,7) (6,8)

(6,4) (1,4)

Determine connected components of this graph using stepwise simulation of mesh algorithm

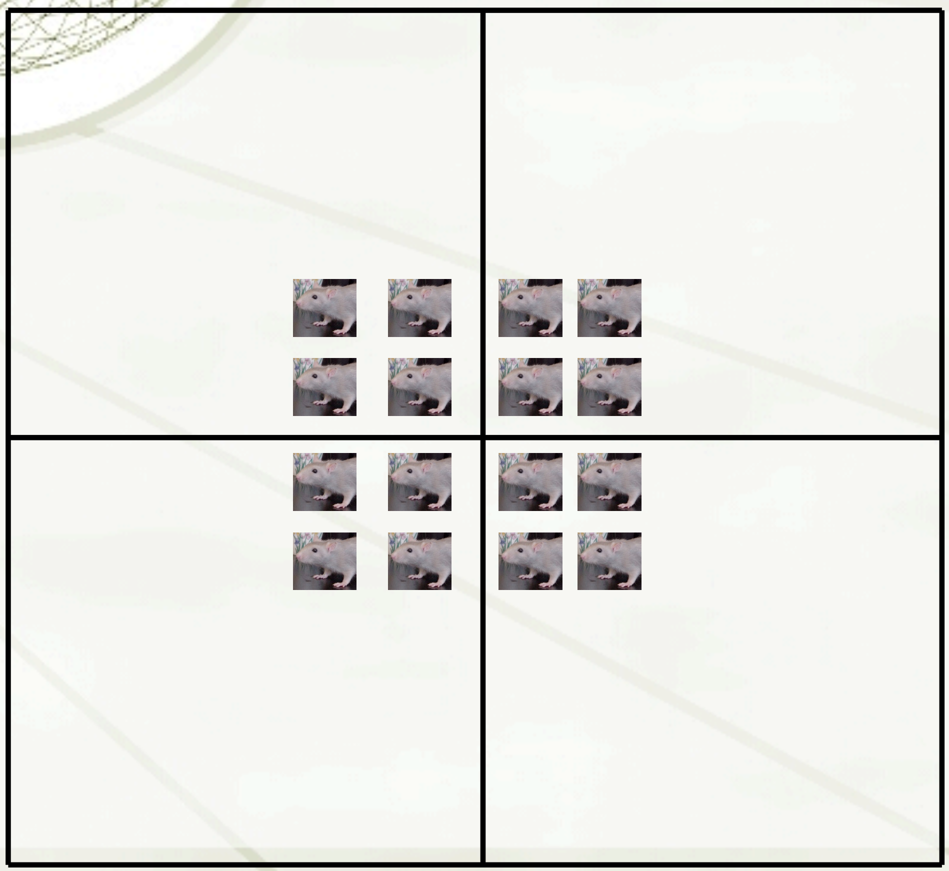
Recursive Stages: Collect Edge Info

Relative to previous stage:

4x rats/square
2x edges, move
2x distance

thus same time

Recursive Stages: Stepwise Simulate Mesh Algorithm



Relative to previous stage:

mesh:
√2x edgelength
√2x time
√8x power

simulation:
1/√2x time

Time

Standard mesh: peak power n, time $\theta(\surd n)$

r rats, i.e., peak power r

- Initial labeling: $\theta(n/r)$
- Merging regions: $\log_4(n/r)$ levels, each $\theta(n/r)$, total $\theta(n \log n / r)$

- Thus with peak power only $\surd n$, time $\theta(\surd n \log n)$: nearly perfect speedup, nearly minimal mesh time
- Can the extra log term be eliminated?

Additional Results

- Similar approaches, yielding similar power reductions and times, for problems such as

  - For each component in the image, find a nearest neighbor and the distance to it

  - Given the adjacency matrix of a graph, label the connected components and find a minimal spanning forest

Sample Data Movement Operation

- Suppose just want to find top of each figure

- map-reduce: every proc creates record (figure label,y-value,proc_coord)
- Sort by label: map
- Find max within figure's interval (reduce), add to record
- Sort on proc_coord, sending record back to original proc
- Total data movement: $\theta(n^{3/2})$: Power Hungry

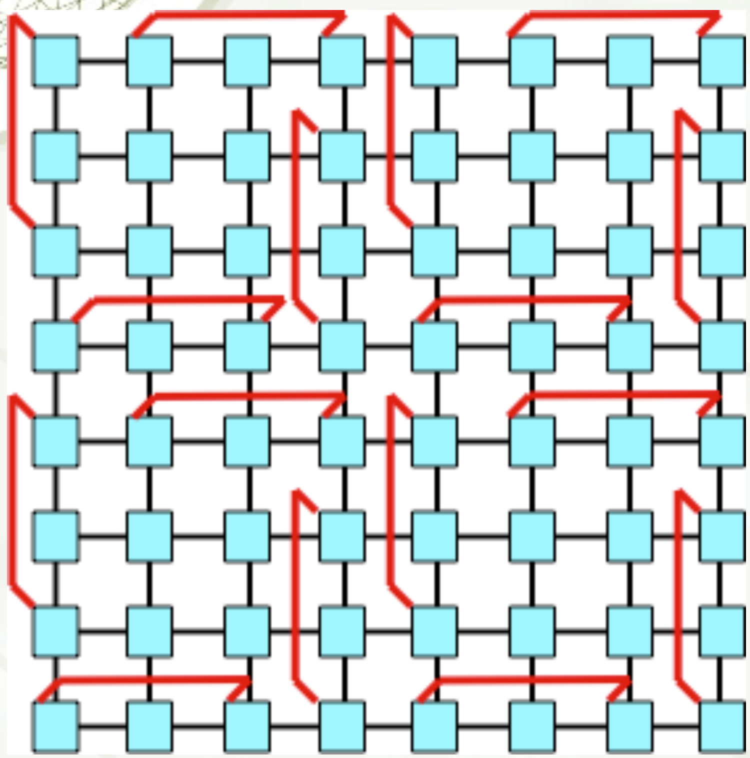
- Power-lite: labeling approach, movement $\theta(n \log n)$

- Note: map-reduce used by Google, Yahoo

Some Research Directions

- Figure out how to stop people from being squeamish about rat algorithms
- Expand (currently small) set of power-constrained algorithms, characterize lower bounds, are there competitive algorithms, etc.
- Develop appropriate power model(s) for reconfigurable mesh

What if Space is actually Matter (Computers) + Light?



Optics: distance not as important: wormholes

Theory grounded in reality: experimental chips with optical waveguides

Mesh + single optic layer can achieve $\Theta(\log n)$ comm diameter

Basic Open Questions, Mesh + Optics

- How should the optics be layed out if have
  - only 1 layer, i.e., cannot cross
  - only 2 layers
  - computer is 3-dimensional
- Which problems can be solved
  - faster?
  - with less energy?
- What should they be called? Opmesh? Mesh +op (“meshpop”)?

References

- Material about BlueGene, ZebraNet, Earth Simulator, Game of Life, easily findable on web
- *Theory of Self-Reproducing Automata*, by John von Neumann, edited and completed by A.W. Burks, Univ. llinois Press, 1966.
- *A New Kind of Science*, S. Wolfram, Wolfram Media, 2002. Beware of ego.
- Collected mesh algorithms (and some for cellular automata), DMOs, and references: *Parallel Algorithms*: *Meshes and Pyramids*, R. Miller and Q.F. Stout, MIT Press, 1996
- Constant-time parity: "Parallel computations on reconfigurable meshes'', R. Miller, V.K. Prasanna Kumar, D. Reisis and Q.F. Stout, *IEEE Trans. Computers* 42 (1993), pp. 678-692
- Rat logo: www.ratfanclub.org
- Maze: en.wikipedia.org/wiki/Maze
- Power-constrained algorithms: contact author
- Mesh + optics: brand new area, no papers yet