Cellular Computing on a Linux Cluster

Alexei Agueev, Bernd Däne, Wolfgang Fengler

TU Ilmenau, Department of Computer Architecture

Topics

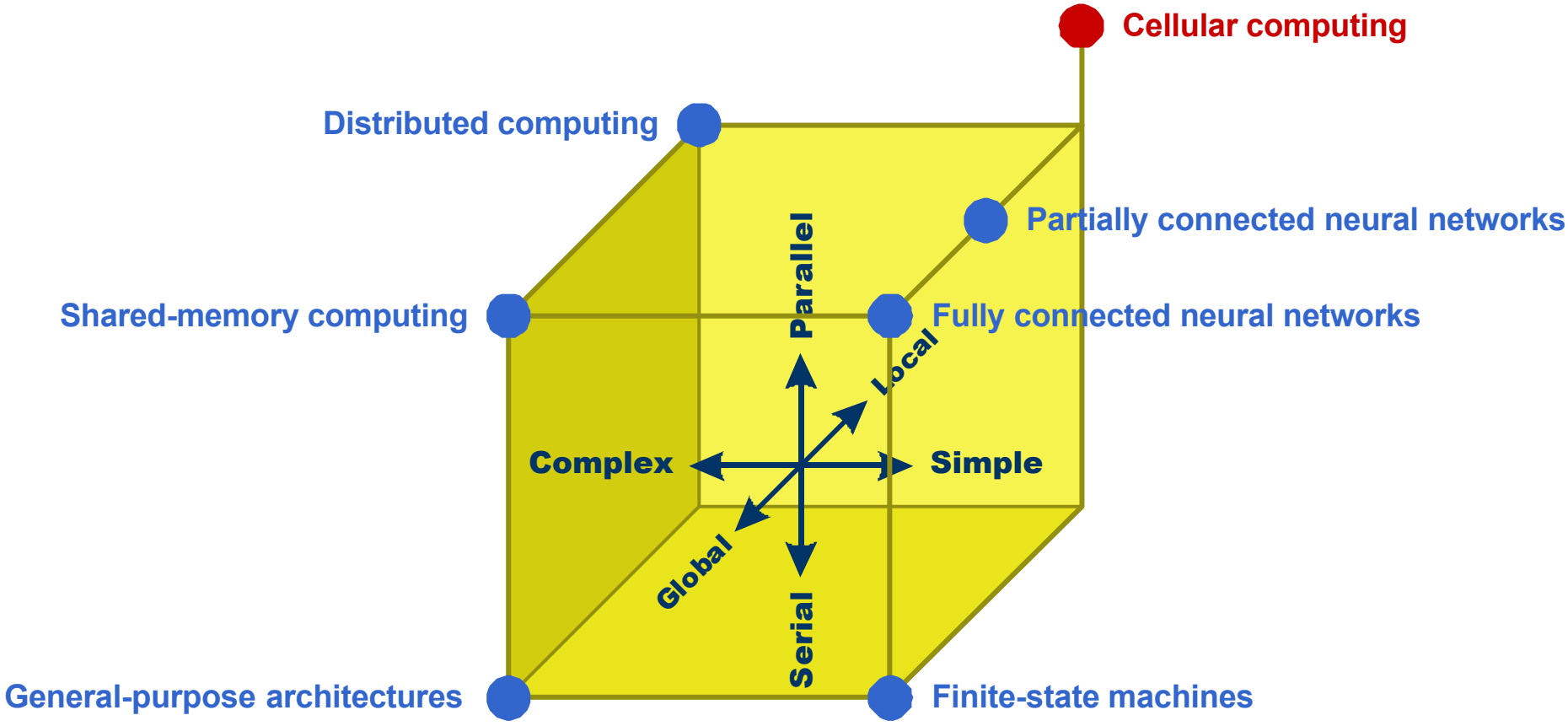
1. Cellular Computing
2. The Experiment
3. Experimental Results
4. Conclusion

1. Cellular Computing

- Extension of cellular automata:
  n-dimensional regular grid of connected cells
- Each cell:
  - State
  - Algorithm
- Major types:
  - Synchronous vs. asynchronous
  - Uniform vs. non-uniform

„cellular computing = simplicity + vast parallelism + locality“ (Sipper)

Sipper‘s Scheme



Moshe Sipper: The Emergence of Cellular Computing. in: IEEE Computer, July 1999, pp. 18-26

Benefits and Examples

- Benefits:
  - Scalability
  - Robustness
  - Simple approach to parallel programming
- Examples:
  - Image processing
  - Pseudorandom numbers
  - Optimizations

Cellular Computers

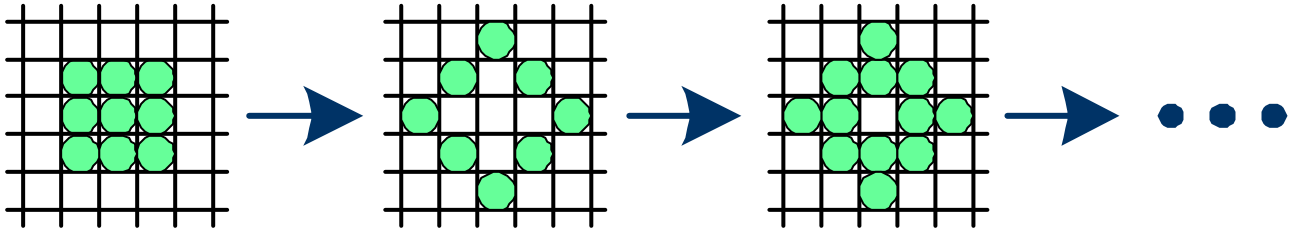
- Real cellular computer:
  - Direct hardware implementation
  - Highly homogenous chip structure
  - Algorithm fixed or loadable
- Virtual cellular computer:
  - Simulation of a cellular structure
  - Runs on single processor or multiprocessor
  - Algorithm loadable

2. The Experiment

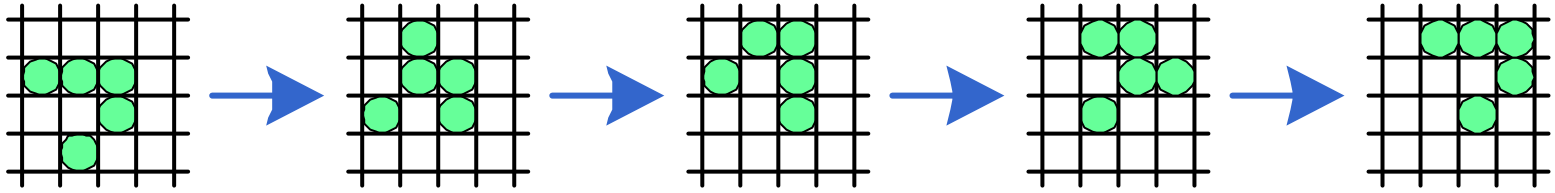
- Virtual cellular computer,
  implemented on a workstation cluster
- Parts:
  - Distributed implementation of a virtual cellular computer
  - Benchmark application: Conway‘s „Game of Life“
- Questions:
  - Performance benefits from coarse-grain parallelism
  - Cellular computing as approach to parallel programming for non-cellular distributed architectures

Conway's Game of Life: Rules

- A living cell with 0 or 1 neighbours dies from isolation.
- A living cell with 4 or more neighbours dies from overcrowding.
- A dead cell with exactly 3 neighbours becomes alive.
- All other cells remain unchanged.



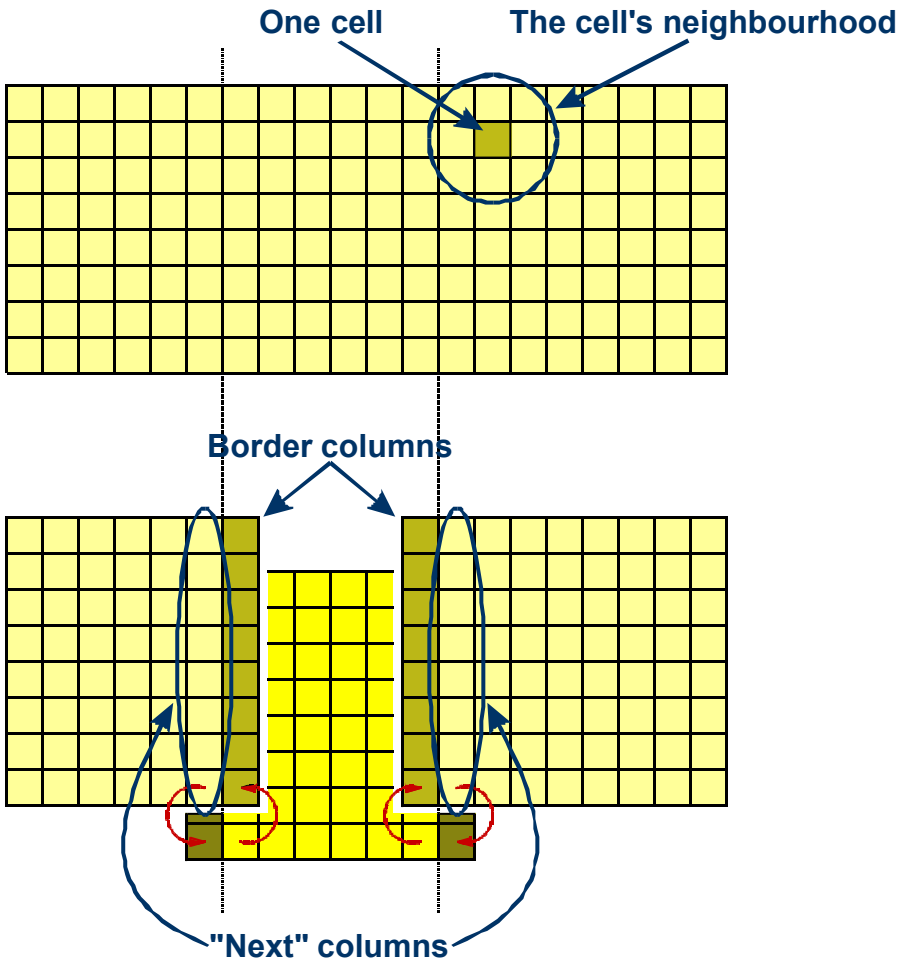
Conway‘s Game of Life: Sample Patterns



Distributed Implementation

- Communicating by message passing
- Cutting the cellular field into equal parts
- Correcting border columns by communicating results of overlapping parts
- **1** master node - **n** slave nodes

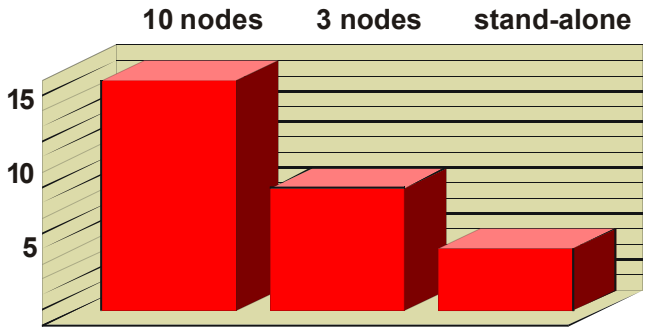
Cutting the Field



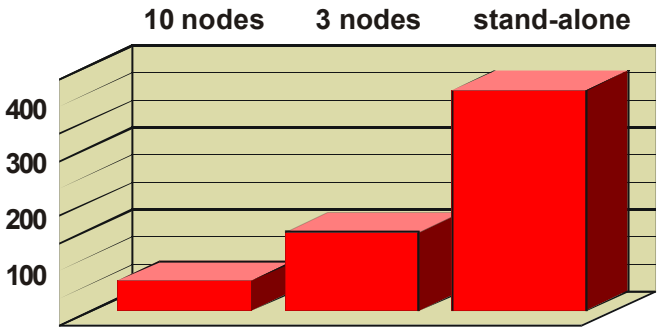
Technical Detail

- 11 node PCs (PIII/500, 512Mb)
- Linux OS
- Gigabit Ethernet network, optical media (star topology, fully switched)
- MPI middleware (lam 6.2b)

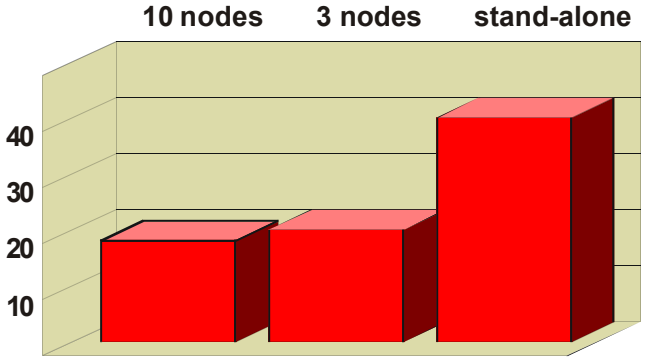
3. Experimental Results



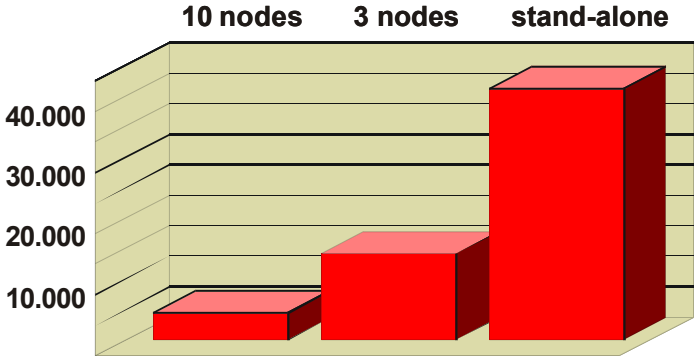
a) field size: 100 x 10 cells



c) field size: 1.000 x 100 cells



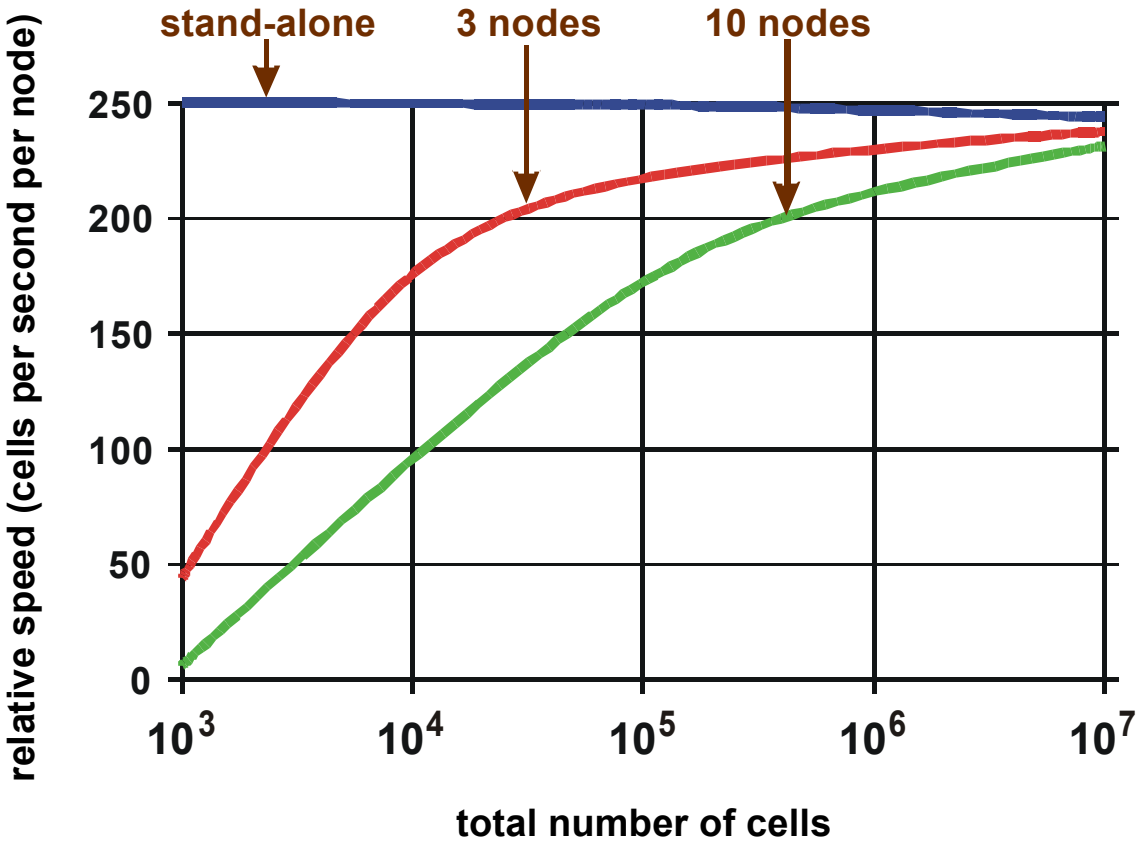
b) field size: 100 x 100 cells



d) field size: 10.000 x 1.000 cells

Runtimes in seconds, for 10.000 iterations

Relative Performance



4. Conclusion

- Distributed implementation works
- For large fields speedup approachs ideal values
- Overhead comes from OS functions rather then node communication:
  - communication amount is proportional to number of rows
  - but: overhead per row proves to decrease when number of rows is increasing

Further Work

- Generalize from 2-dimensional to n-dimensional
- Universal application interface
- Benchmarking more applications
- Comparing to non-cellular distributed solutions of same problems