Pursue robust indefinite scalability

David H. Ackley
Daniel C. Cannon
University of New Mexico
Computer Science

With Lance Williams & Thomas B. Jones
and the Robust Physical Computation Group

HOTOS XIII
Napa Valley, CA
May 9, 2011
Computation must be born again

Instead of being born again, why not just GROW UP?

► OK, our answer might be wrong
Computation's Original Sin

Hardware Shall Be Reliable
Software Shall Be Efficient

► Efficiency and robustness are mortal enemies
A path to redemption: 
*Indefinite scalability*

- **Mantra:** Let space be space, let time be time
- **Sacrificing:**
  - ✗ Fixed-width addresses, unique node names.
  - ✗ Logarithmic global communication cost
  - ✗ Clock and phase synchronization
- **Embracing:**
  - ✔ Opportunistic reproduction for ||ism & robustness
  - ✔ Movability for configuration, manifest destiny, ...
  - ✔ Multilevel robustness: Up to the end-user
Strategy for universe domination

- Take $10^{-14}$ hit to move to hosted physics; using small von Neumann machines as the 'ether'
- Hosted processor/memory/communication are assigned to regions of space at run-time
- You as god: *It's the expressiveness, stupid!*
- Optimize 'periodic table', buy back $10^x$ for $x=?$
- Deploy at scale.
Example: Software engineering as artificial chemistry

The Movable Feast Machine

spatial computing
relative addressing
local connectivity
indeinitely scalable
fixed size atoms/sites for mobility
parallel asynchronous update
element-oriented programming

/* DReg: Dynamic Regulator. */
element DReg() = 0xdba {
    if n:anyAt(1), n is Empty, odds(1,1000) then n = DReg;
    if n is Empty, odds(1,200) then n = Res;
    if n is DReg, odds(1,10) then n = Empty; // limit DRegs
    if odds(1,100) then n = Empty;
}
Demon Horde Sorting: Robust Computation Example

- **Task:** Flow sort endless data stream
- 'Maxwell's Demon' sorting elements stabilized by BC+DReg
- **Surprise:** Sorting quality vs data rate..