Computational Emergence Research Area

Richard P. Gabriel
IBM Research

Ultra Large Scale Systems

- diverse components
- ephemerality
- longevity
- continual change
- continual failure
- inconsistent
- beyond human comprehension

Prescription vs Emergence

- not all aspects of the ULS system will be programmed via prescription
- computational emergence:

Macroscopic characteristics and behavior created by the interaction of simple agents and entities obeying simple, local rules sometimes mediated by artifacts.

Prescription vs Emergence

- (computational) emergence…
  - …will be used to incentivize, constrain, entice, and influence…
  - …both the software and the development of the system

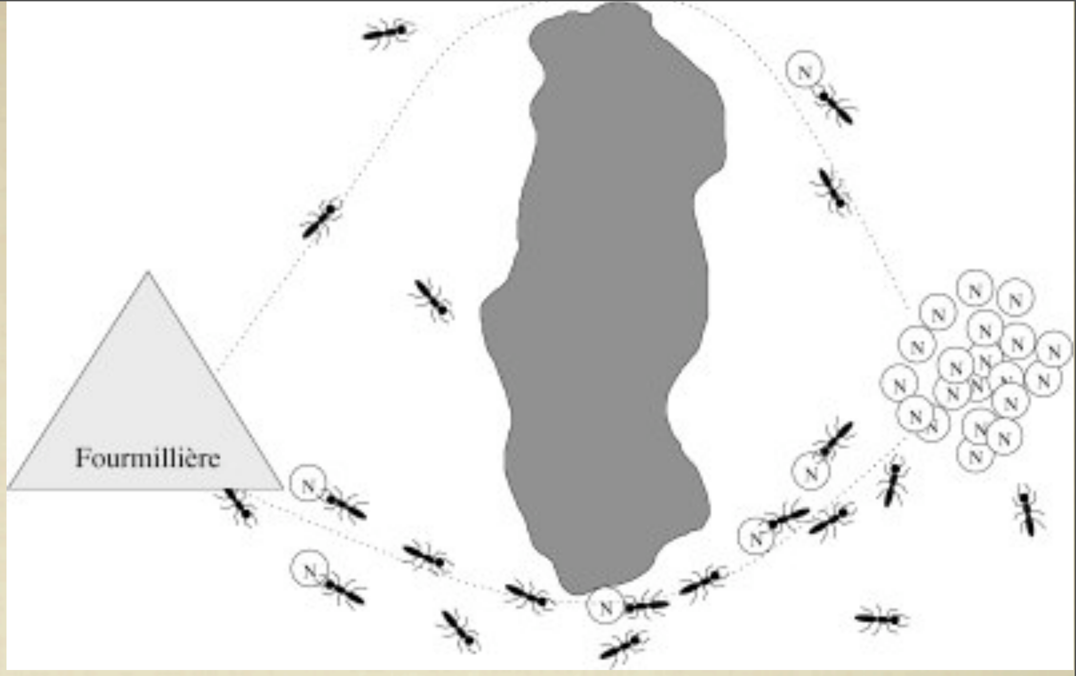
Mechanism Design

- game theory & microeconomics
- ...for designing & implementing the ULS itself
- ...to use scarce resources by using agents
- ...used for self-adaptation by having mechanisms that achieve an end be designed by the system itself

Metaheuristics

- ant-colony optimization (finding short paths)
- simulated annealing (search avoiding local minima)
- digital evolution / genetic algorithms (novel / thorough search of design spaces)
- swarm intelligence (local rules creating global effects)

Metaheuristics

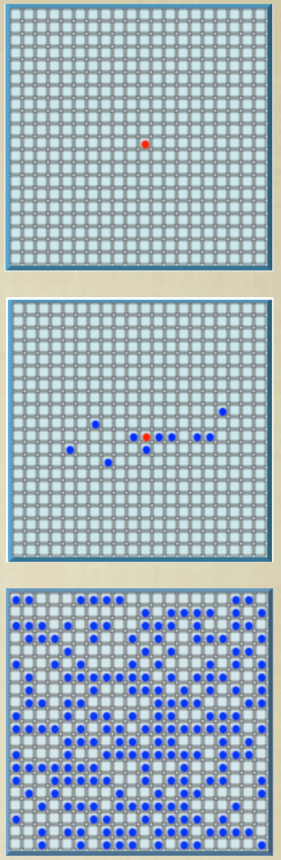


- ant-colony optimization (finding short paths)
- simulated annealing (search avoiding local minima)
- digital evolution / genetic algorithms (novel / thorough search of design spaces)
- swarm intelligence (local rules creating global effects)

Simple Examples

- distributing computations over a multicore cpu chip using a gas diffusion model
- moving collaborating objects closer and non-collaborators farther apart using attraction & repulsion in a potential field

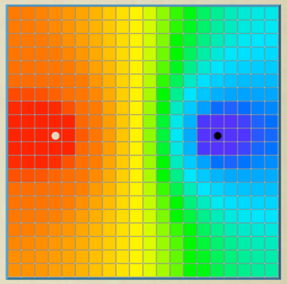
Simple Examples



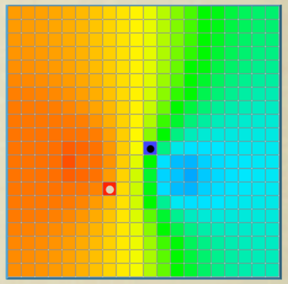
- distributing computations over a multicore cpu chip using a gas diffusion model
- moving collaborating objects closer and non-collaborators farther apart using attraction & repulsion in a potential field

Simple Examples

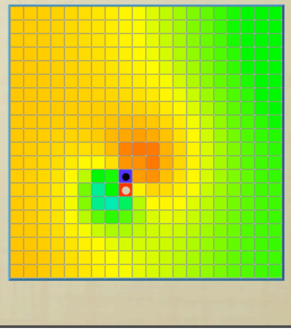
- distributing computations over a multicore cpu chip using a gas diffusion model
- moving collaborating objects closer and non-collaborators farther apart using attraction & repulsion in a potential field



After three steps:



After three more steps:



Other Techniques

- Bayesian techniques (Recovery-Oriented Programming, Armando Fox / Dave Patterson for Amazon.com, e.g.)
- machine learning (system tuning, e.g.)

Exotic Ideas

- evolving designs through crossover
- evolving resource-sharing policies
- evolving modularizations
- creating intercomponent glue
- completing the details of an implementation

More Exotic Ideas

- biologically inspired mechanisms for self-healing and self-adaptation (immunology, e.g.)
- moving toward *autopoietic* systems
  - through embedding
  - through forms of micro-reboots

Autopoiesis

“systems that are defined as unities, as networks of productions of components, that recursively through their interactions, generate and realize the network that produces them and constitute, in the space in which they exist, the boundaries of the network as components that participate in the realization of the network”

–Humberto R. Maturana, “Autopoiesis” 1981

Autopoiesis

"In such a natural product as this every part is thought as owing its presence to the agency of all the remaining parts and also as existing for the sake of the others and of the whole . . . the part must be an organ producing the other parts, each consequently reciprocally producing the others."

–Immanuel Kant, "The Critique of Judgment" 1790

Autopoietic Approaches

- helping keep the system alive and running well
- helping keep the system running correctly
- using multicore machines to supply the horsepower + a sense of (physical) locality

Computational Emergence Research Area

Richard P. Gabriel
IBM Research

Computational Emergence Research Area

Richard P. Gabriel
IBM Research