Uniform Distribution of Particles on Cellular Space by Voronoï Diagram

Element for the runtime system of the Blob machine

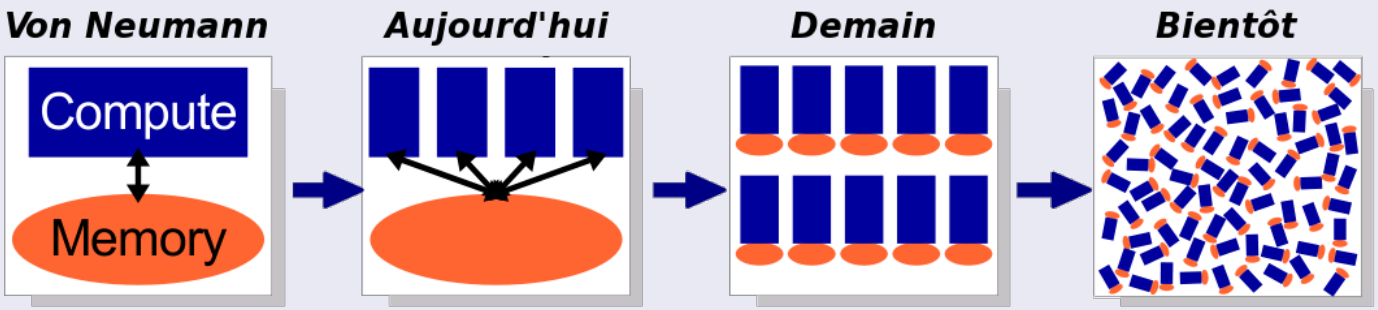
Luidnel Maignan
<luidnel.maignan@gmail.com>

INRIA Futurs Saclay - ALCHEMY Team

Table of Contents

Evolution of computer architectures

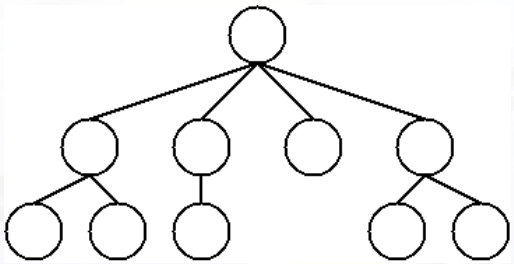
From Von Neumann to Computing Medium



Blob Computing

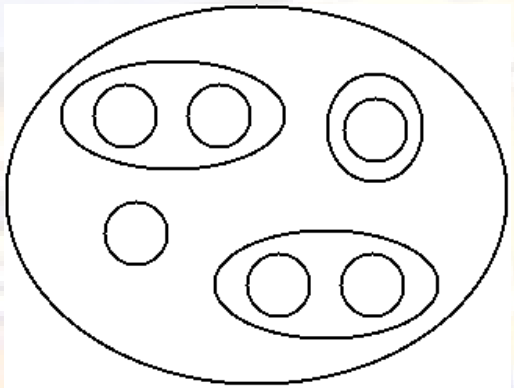
- Goal: To organize the computing medium to allow programmability
- How? Virtual Machine offering the vision of a Self-Developing Automata Network

Self-Developing Automata Network in Blob Computing



Blob Self-Developing Automata Network

- very small computing node to allow mapping on arbitrary grain
- In simple cases, the network is a tree



Mapping on 2D space

- Each parent encompasses its children
- Communications require nodes to be close
- Creations of new nodes require existing nodes to be far

Physical Nature of Blobs

- Communications and node creations have opposed needs
- Nodes move to optimize their placement
- Attraction/Repulsion

- Repulsion leads nodes to fill the space
- Attraction leads nodes to limit the used space
- Attraction/Repulsion $\Rightarrow$ Uniform distribution/Bounding distribution area

Problem Presentation

Framework

- Computing Medium = Cellular Automaton
- Nodes have no child
- Fine grain $\Rightarrow$ a node = a cell: particles

Challenge

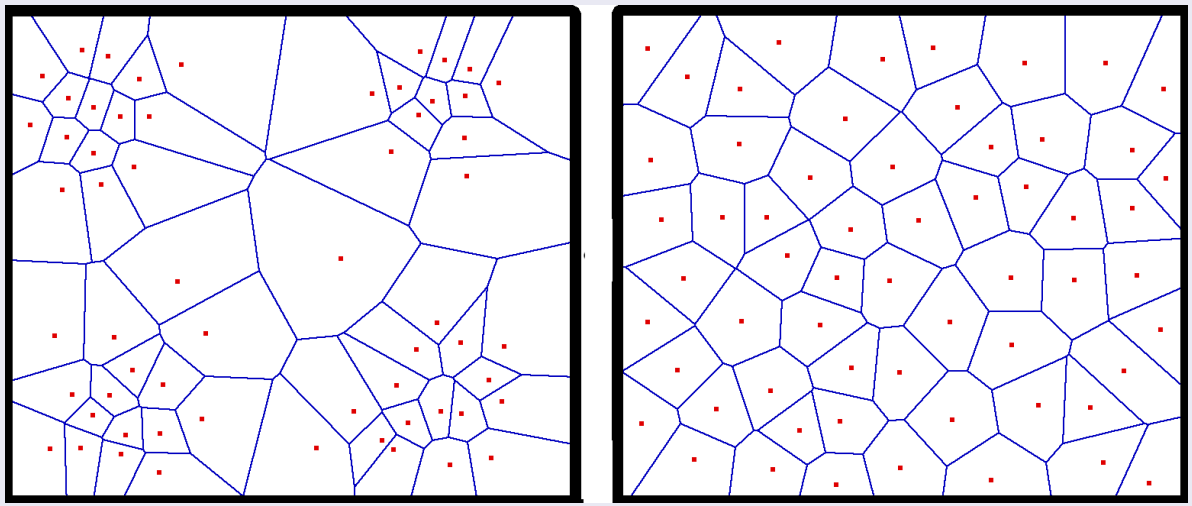
To move particles to get a uniform distribution in the cellular space, by using only local rules

State of the art

- *J. Hardy, O. de Pazzis, and Y. Pomeau (1976): HPP*
  - Obtain gas properties with a cellular automaton
  - $\Rightarrow$ distribute their particles uniformly in the cellular space
- *Gruau et Moszkowsky (2003): The Blob Division*
  - Re-use HPP rules to implement a Blob primitive
  - Problem:
    - Particles never stop moving and computing their movement
    - HPP rules work only for particles, and not for bigger entities *(realistic case)*

Intuition

- How people distribute themselves uniformly in a room ?
- $\Rightarrow$ Each person try to stay as far as possible from other people, and then go to the center of the free space around him
- Formalization: centering in his Voronoï region



General Algorithm

Repeat:

1. Compute the Voronoï diagram of the particles
2. Compute the center of each Voronoï region
3. Move each particle to the center of its region

Until stabilization

Cellular Algorithm

Repeat:

1. Compute the Voronoï diagram of the particles
   - Compute distances to the nearest particle of each cell
   - Detect cells belonging to the Voronoï diagram
2. Compute the path to the center of each Voronoï region
3. Move each particle to the center of its region

Until stabilization

Voronoï Diagram

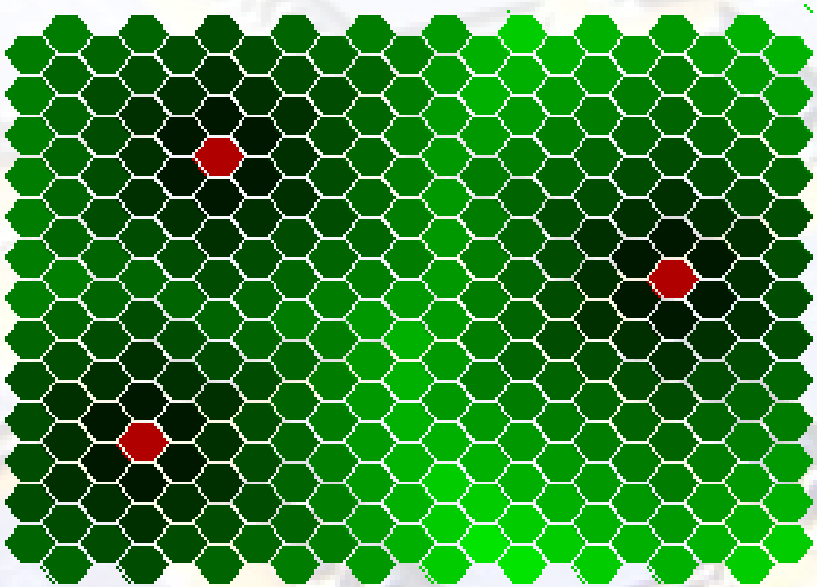
Distances to the nearest particle

$$d_i(t+1) = \begin{cases} 0 \text{ if particle} \\ 1 + \min_{j \in N_i} d_j(t) \end{cases}$$

Detect cells belonging to the Voronoï diagram

Detection of critical points:

- local maxima
- saddle points



Voronoï Diagram

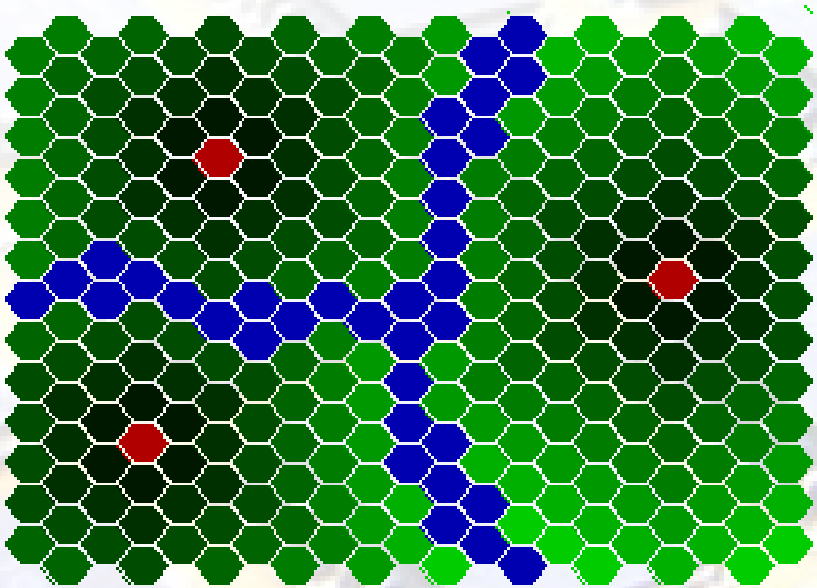
Distances to the nearest particle

$$d_i(t+1) = \begin{cases} 0 \text{ if particle} \\ 1 + \min_{j \in N_i} d_j(t) \end{cases}$$

Detect cells belonging to the Voronoï diagram

Detection of critical points:

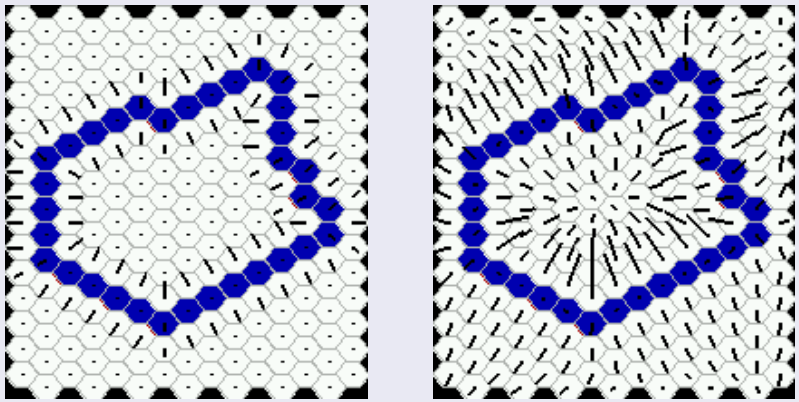
- local maxima
- saddle points



Movement to the center

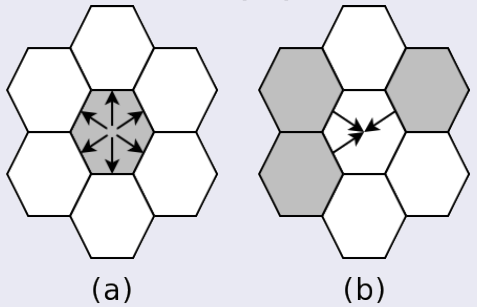
Centering vectors

$$\vec{c_i}(t+1) = \frac{1}{|N_i|} \sum_{j \in N_i} \vec{c_j}(t)$$

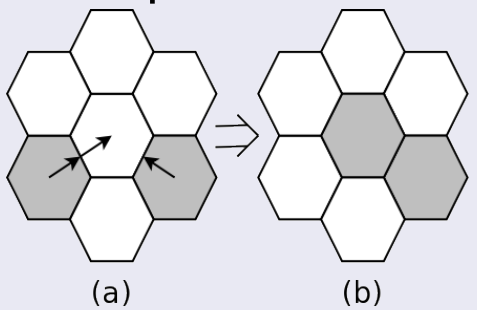


Move without particle loss

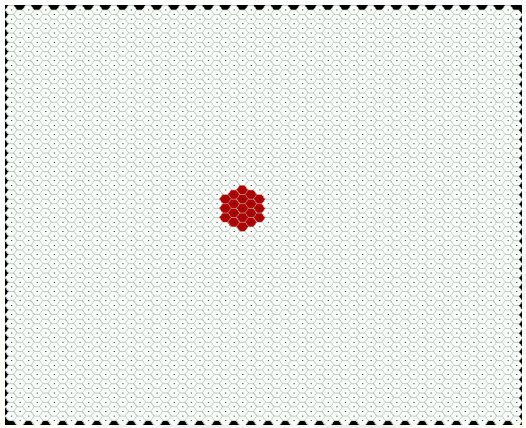
- Possibilities for: (a) a particle, (b) a empty cell



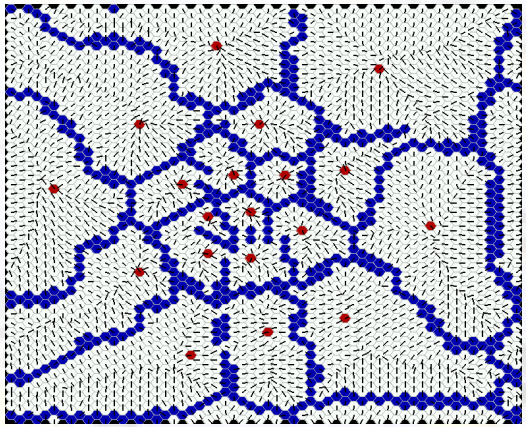
- Exemple:



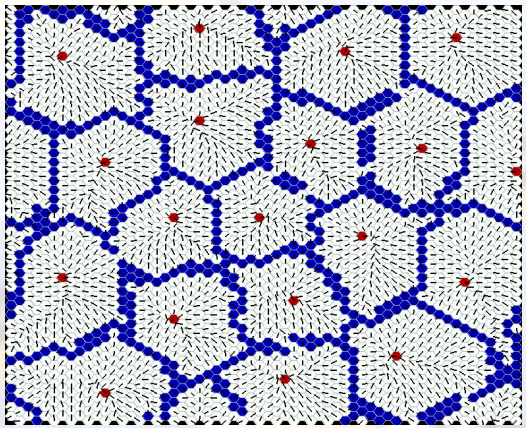
Experimentation



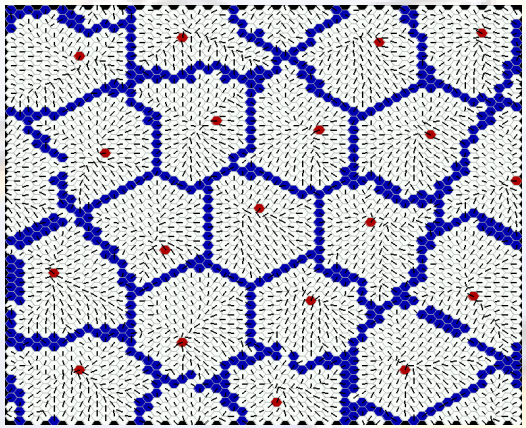
Initial configuration



after 50 iterations



after 150 iterations



after 250 iterations

- Attraction: adding membranes to bound the distribution area
- Apply the same algorithm on membranes to manage a full tree structure