

Multi-Agents Systems on HPC

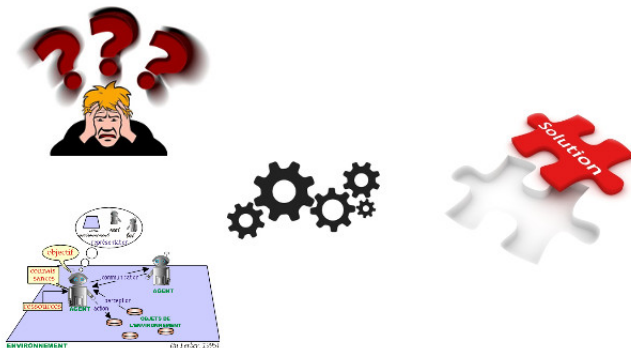
Luxdem Team - PCOG Meeting

Alban ROUSSET

Academic background

- Bachelor degree : University of France-Comte
- Master degree (Parallel and distributed systems) : University of Franche-Comte
- Phd degree : Femto-st/CNRS institut
- Research associate : University of Luxembourg (LuxDem Team)

Multi-Agents System (MAS)



Multi-Agents System (MAS) :

- A paradigm to model problems
- Based on interactions between entities called agents

The increasing of models size

Increasing the size of models implies limitations

- Centralized systems are often no longer sufficient
 - memory, power of computation, ...



One solution

- The use of parallel and distributed systems
 - cluster, grid, ...

The increasing of models size

Increasing the size of models implies limitations

- Centralized systems are often no longer sufficient
 - memory, power of computation, ...



One solution

- The use of parallel and distributed systems
 - cluster, grid, ...

Parallel and Distributed MAS

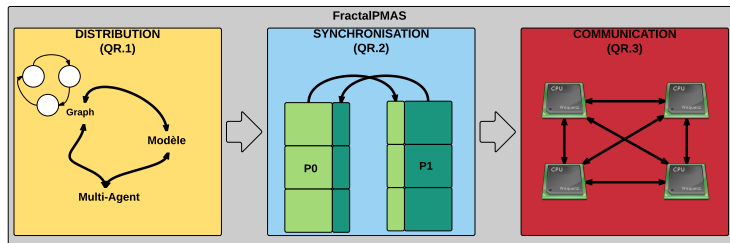


Parallel systems implies :

- Divide the problem in different parts
- Assign each part to a processus

Contributions

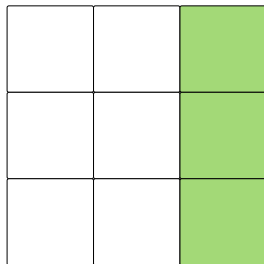
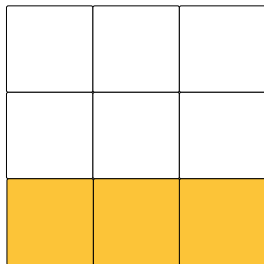
- Distribute and Model a MAS
- Evaluate synchronisation policies in MAS
- Communication schema for MAS



Limits of grid distribution in PDMAS

Limits of grid distribution in PDMAS :

- divide in line or column



Lack of flexibility in the distribution.

Classical model : Prey-Predator

Goal : Exploring the stability of ecosystems

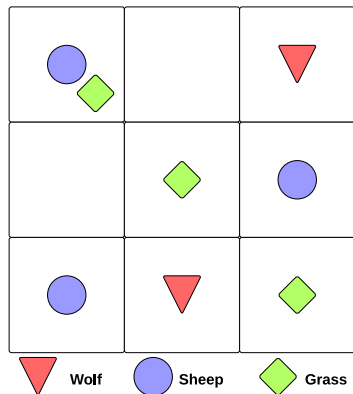
Model is composed of 3 types of agents : Wolf (predator), Sheep (Prey) and Grass (Prey).

Each agent is composed of 4 behaviours :

- move
- food
- die
- reproduce

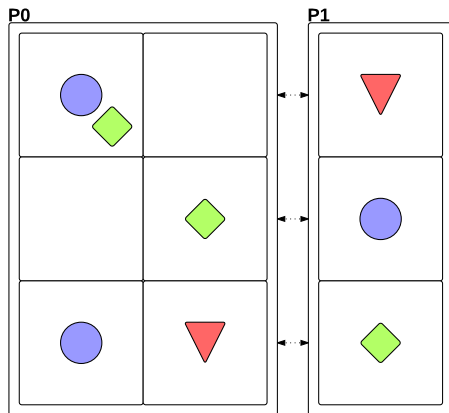
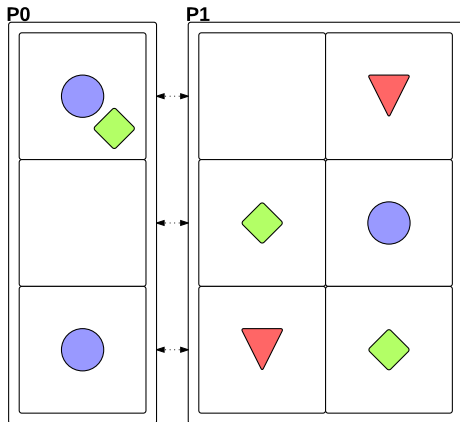
Distribution of Prey-Predator model

Initial configuration of Prey-Predator model based on grid



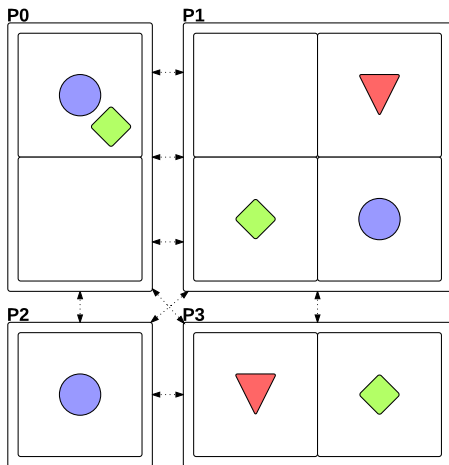
Distribution of Prey-Predator model

On 2 processus



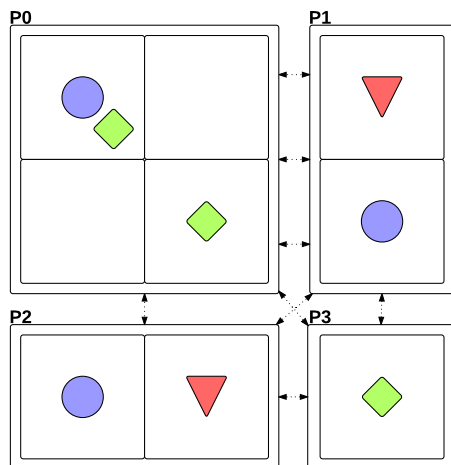
Distribution of Prey-Predator model

On 4 processus



 Wolf
  Sheep
  Grass

◀...▶ Inter-processor communication



 Wolf
  Sheep
  Grass

◀...▶ Inter-processor communication

Nested Graphs formalism

In this formalism, *each entitie* of the model is considered as an Agent :

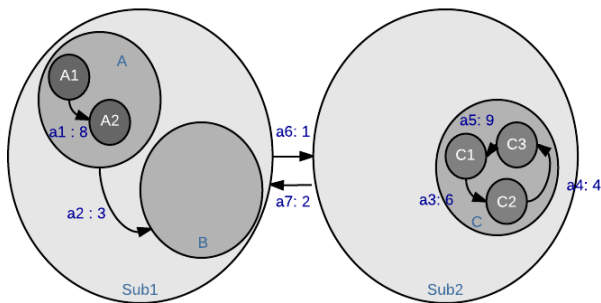
- Agents are represented as Graphs (Agent Graph)
- Agent's behaviours are represented as *Graphs Transformation* (Agent Graph Transformation)
- $\mathcal{G} \in \mathbb{G} \Leftrightarrow \mathcal{G} = \langle \mathcal{G}, \mathcal{A}, \mathcal{T}, \mathcal{L}, \mathcal{V} \rangle$ where $\mathcal{G} \subseteq \mathbb{G}$

Nested Graphs :

- More flexible structure
- Graphical way to model the simulation
- Use of existing framework to distribute the simulation

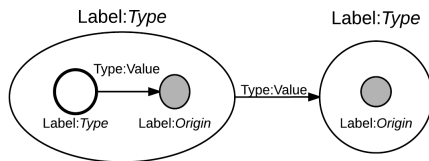
Nested Graphs formalism

Nested Graph Framework :



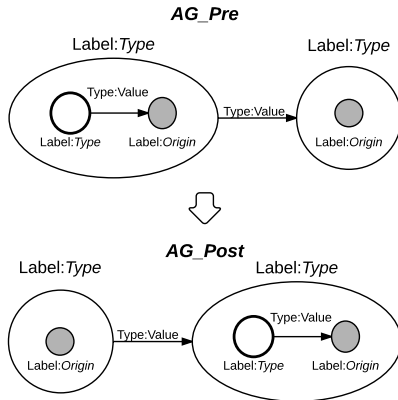
Nested Graphs formalism

What is *Agent Graph* :



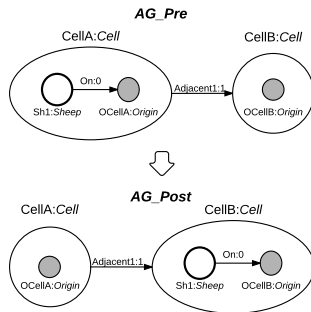
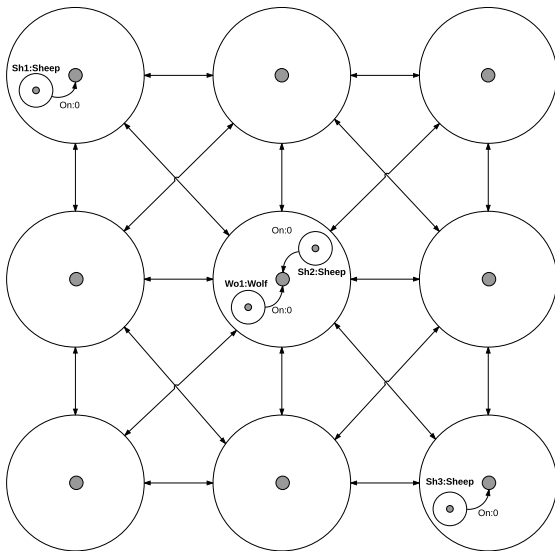
Nested Graphs formalism

What is *Agent Graph Transformation* :



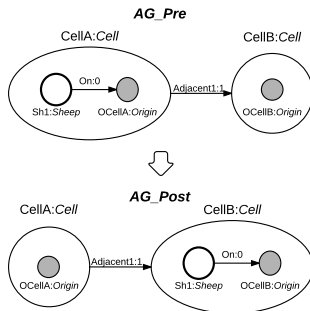
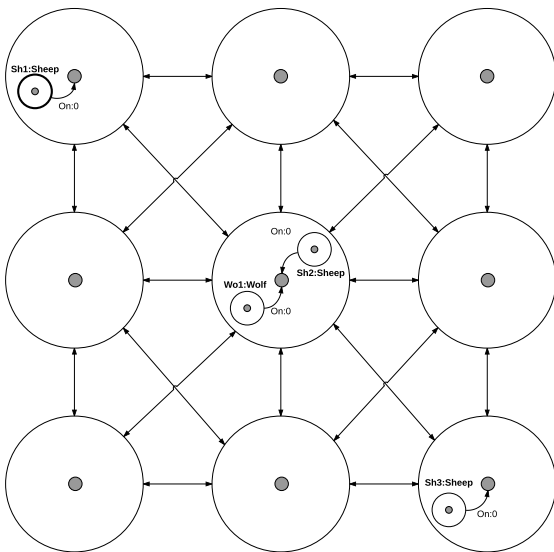
Prey-Predator model using *Nested Graphs*

Move :



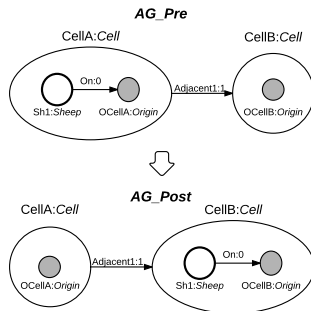
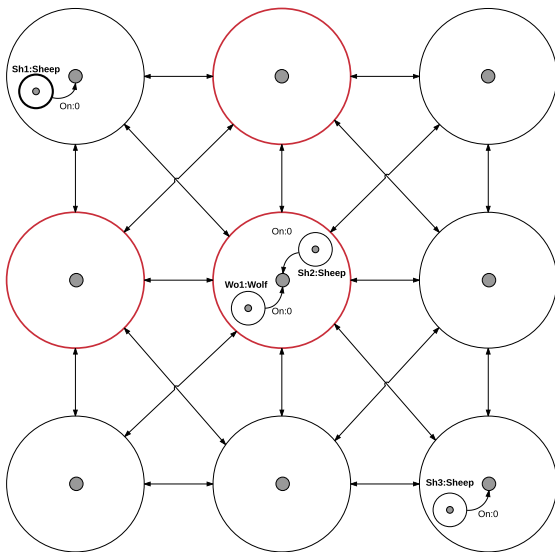
Prey-Predator model using *Nested Graphs*

Move :



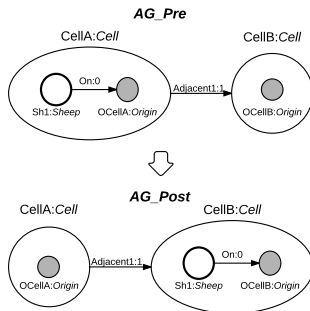
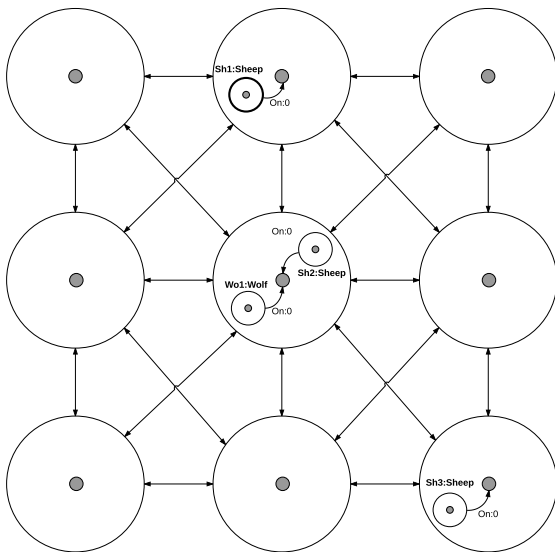
Prey-Predator model using *Nested Graphs*

Move :



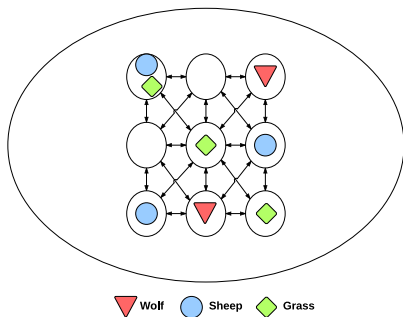
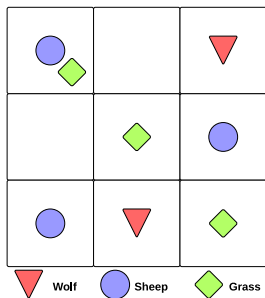
Prey-Predator model using *Nested Graphs*

Move :



Distribution of Prey-Predator model

An initial configuration of the Prey-Predator model using *Nested Graphs*

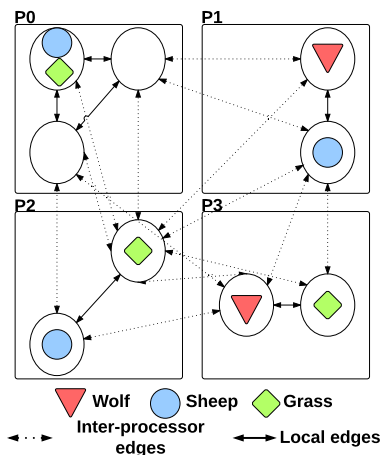
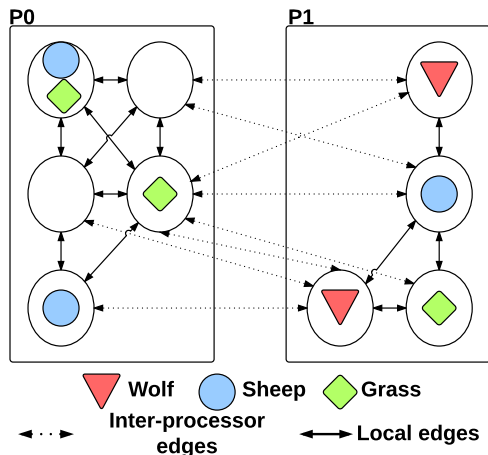


Distribution of Prey-Predator model

how to distribute the graph (simulation) ?

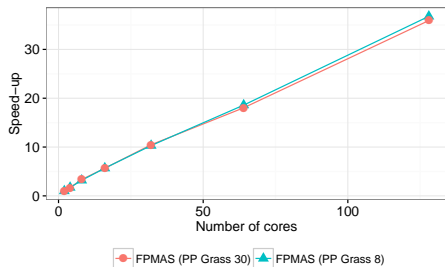
- Use of parallel graph partitioner :
 - Zoltan
 - ParMetis
 - Scotch
- Zoltan allows the use of other partitioners (ParMetis, Scotch)
- Zoltan allows dynamic load balancing

Distribution of Prey-Predator model

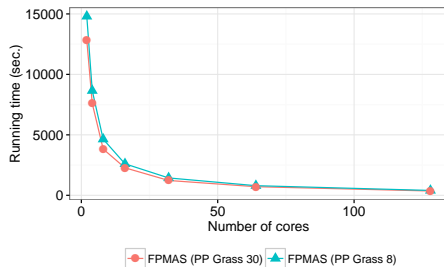


Results : Prey-predator model

FractalPMAS (*Nested Graph*)
from 2 to 128 cores for 2000 timesteps 80000 agents

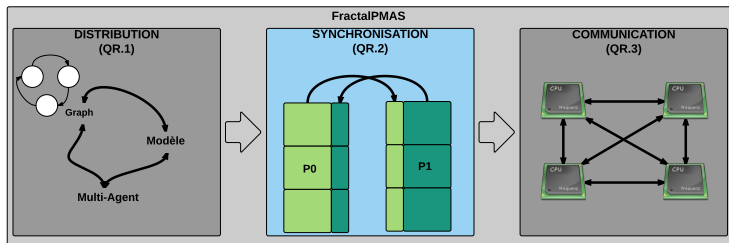


Scalability



Running time

Synchronization



Goals :

- Evaluate impact of synchronization policies on :
 - running time
 - simulation's results

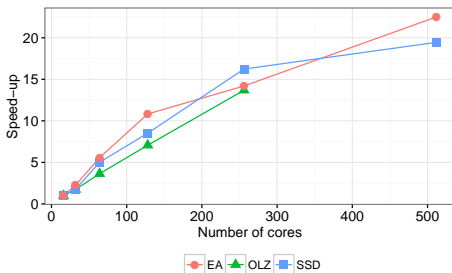
Synchronization policies

Different synchronization policies :

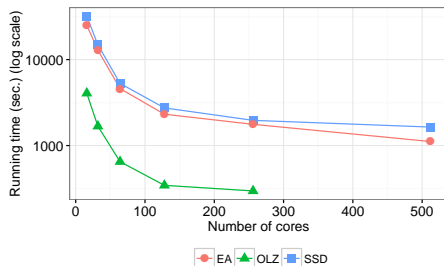
- **No synchronization (NS)**
- OverLapping Zones (OLZ)
- Asynchronous writes (EA)
- **Strong synchronization (SS)**
- Strong synchronization D (SSD)

Impact of synchronization policies

Modèle Virus :



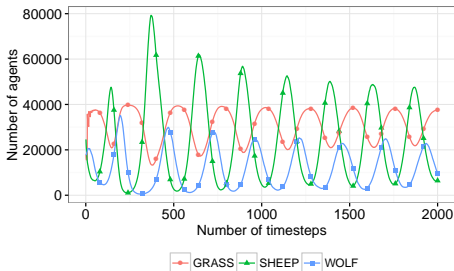
Scalability



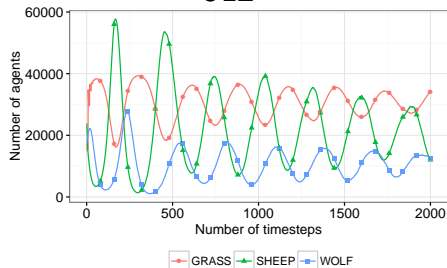
Running time

Prey-Predator : execution results

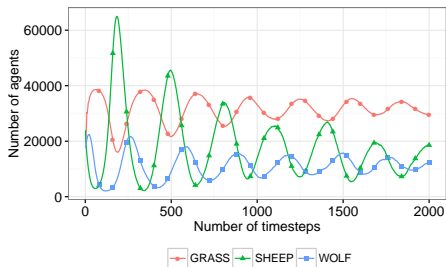
SS



OLZ



SSD



Flocking model : execution results



Merci pour votre attention...

Alban ROUSSET - alban.rousset@uni.lu



Publications	
Journal	2
Int. Conferences	1
Int. Workshops	2
Nat. Conferences	1
Res. Report	2
Total	8