Software for agent based social simulation ~~with raster inputs~~ in distributed HPC environments

Sergiy Gogolenko

Oct 11, WSSP'17

Coe GSS
Centre of excellence for
GLOBAL SYSTEMS SCIENCE

GG bridge. Photo from ©pixabay.com

GSS

scientific evidence to support
policy-making · public action · civic society

HPC

software · platforms

pilots: grean growth, health habits, urban planning } $\oplus \rightarrow$ **LESSONS**

Deliver a stack of **HPC compliant software** for GSS users

- creating realistic synthetic populations
- network reconstruction (creating synthetic social networks, etc)
- parallel and distributed multi-agent systems simulation (for GSS only)
- analytics: basic data and graph analytics, sensitivity analysis, parameter sweep, etc

Agent-based modeling and simulation

Annual Workshops on HPC Forums

- 2012 – **EuroPar**: Workshop on Parallel and Distributed Agent-Based Simulations (**PADABS**)
- 2016 – **IPDPS**: IEEE Workshop on Parallel and Distributed Processing for Computational Social Systems (**ParSocial**)

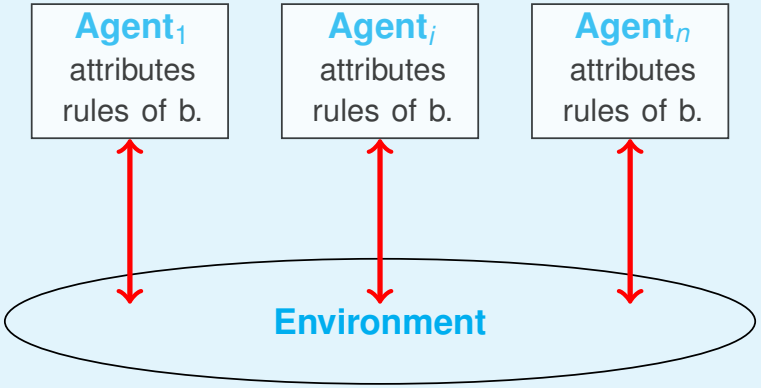
Agent-based modeling and simulation

Annual Workshops on HPC Forums

- 2012 – **EuroPar**: Workshop on Parallel and Distributed Agent-Based Simulations (**PADABS**)
- 2016 – **IPDPS**: IEEE Workshop on Parallel and Distributed Processing for Computational Social Systems (**ParSocial**)

ABMS in a Nutshell

- **bottom-up approach**
  - individual-based simulation → tendency
- **components**
  - agents: discrete autonomous entity
  - environment: GIS, relationships, etc
  - actions and interactions



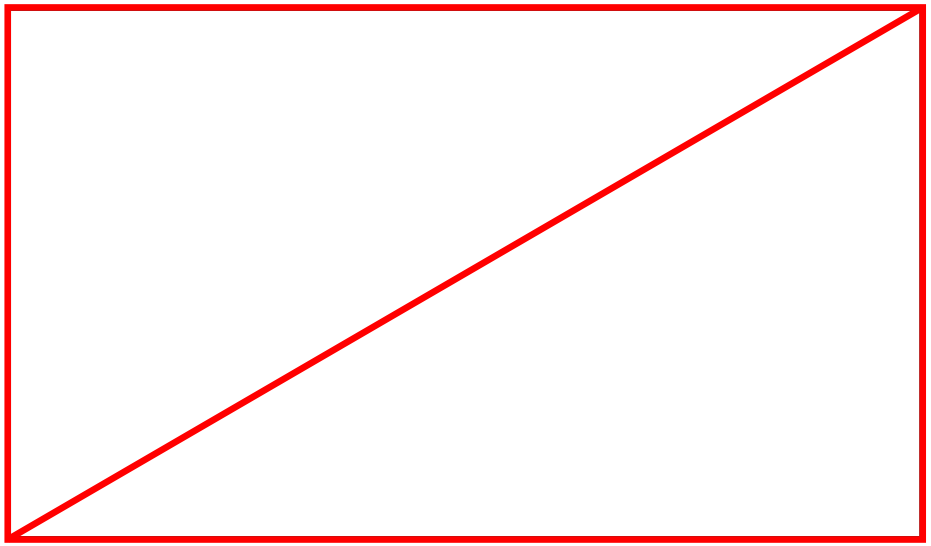
MAS in GSS : Lesson 1. Agents

“molecular dynamics society”

The Simpsons : Treehouse of Horror XIII (Season 27) [from ©fxx.com]

MAS in GSS : Lesson 1. Agents

“molecular dynamics society”



The Simpsons : Treehouse of Horror XIII (Season 27) [from ©fxx.com]

real society is heterogeneous

MAS in GSS : Lesson 1. Agents

- heterogeneous

The Simpsons ©[from ©fxx.com]

MAS in GSS : Lesson 1. Agents

- heterogeneous
- mobile

The Simpsons ©[from ©fxx.com]

MAS in GSS : Lesson 1. Agents

- heterogeneous
- serializable (introspection)
  - I/O
  - data exchange
  - checkpointing
- mobile

The Simpsons ©[from ©fxx.com]

MAS in GSS : Lesson 2. Environment

Border fence Tijuana-San Diego. Photo from ©wikimedia.org

MAS in GSS : Lesson 2. Environment

Border fence Tijuana-San Diego. Photo from ©wikimedia.org

MAS in GSS : Lesson 2. Environment

Border fence Tijuana-San Diego. Photo from ©wikimedia.org

MAS in GSS : Lesson 2. Environment

Border fence Tijuana-San Diego. Photo from ©wikimedia.org

MAS in GSS : Lesson 2. Environment


Earth at Night. Photo by NASA ©wikimedia.org

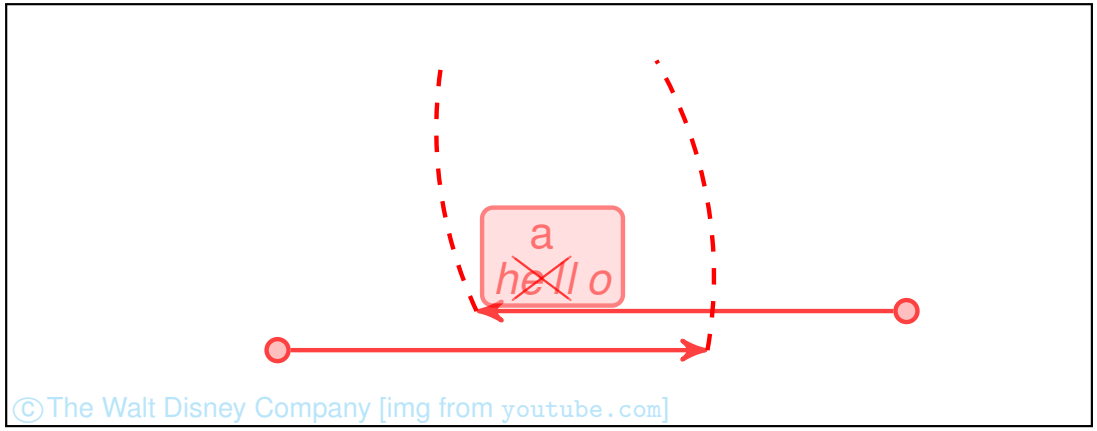
MAS in GSS : Lesson 2. Environment

MAS in GSS : Lesson 3. Interactions

- **short distance** (spatial proximity)

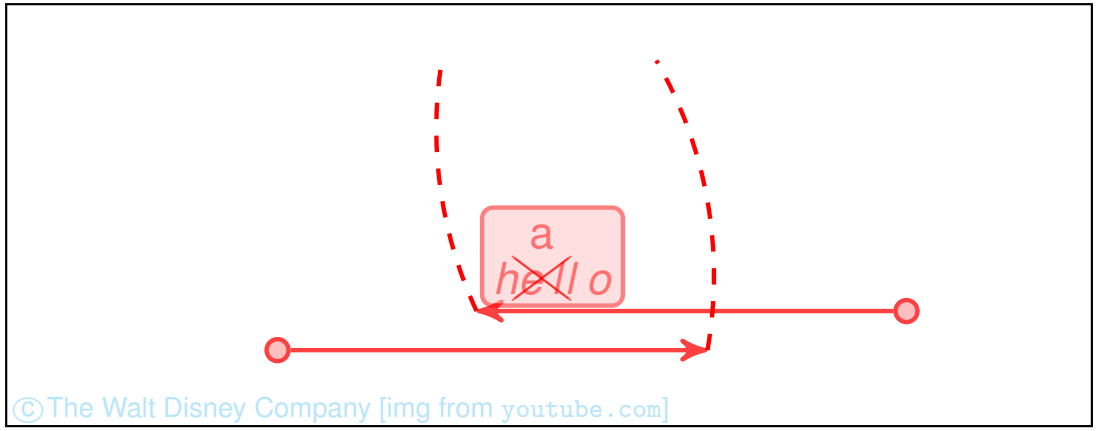
MAS in GSS : Lesson 3. Interactions

- **short distance** (spatial proximity)
  - metric: *halo* (area of interest – AOI)
  - *halo* size: agent type independent
  - BC: von Neumann, Moore, ~~periodic~~



MAS in GSS : Lesson 3. Interactions

- **short distance** (spatial proximity)
  - metric: *halo* (area of interest – AOI)
  - *halo* size: agent type independent
  - BC: von Neumann, Moore, ~~periodic~~



- **long distance** (social relationships)
  - undirected graphs
    - dynamic (evolving)
    - might be weighted
  - might need several graphs

MAS in GSS : Lesson 3. Interactions

scientific collaboration network (©Olivier H. Beauchesne)

MAS in GSS : Lesson 3. Interactions

scientific collaboration network (©Olivier H. Beauchesne)

social relationships may dominate

MAS in GSS : Lesson 4. Users

Technical ~~traits~~ background of average GSS modeler

- `Python` and/or `R`
- minimum boilerplates
- ~~spend time on mastering low level parallel APIs (MPI, OpenMP, CUDA,...)~~

MAS in GSS : Lesson 4. Users

Technical ~~traits~~ background of average GSS modeler

- `Python` and/or `R`
- minimum boilerplates
- ~~spend time on mastering low level parallel APIs (MPI, OpenMP, CUDA,...)~~

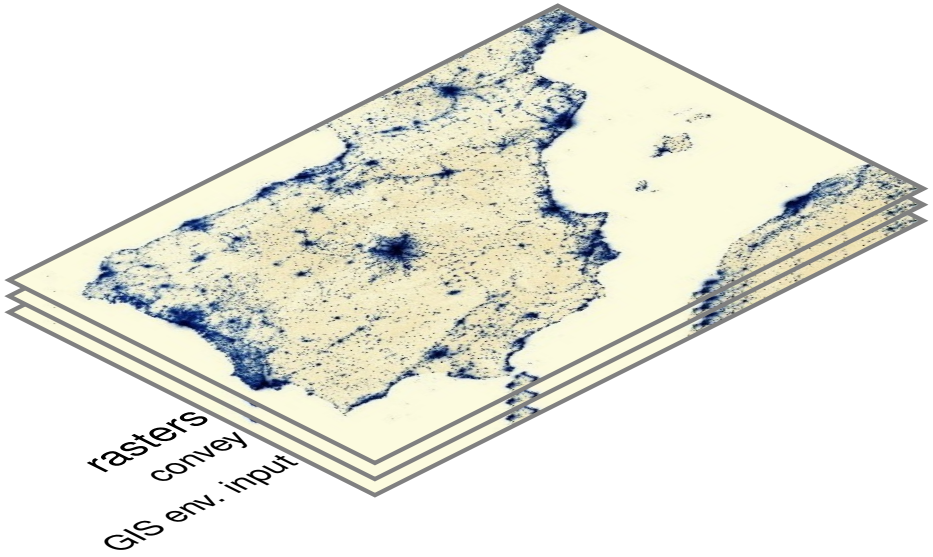
Embeded DSLs

- linear algebra

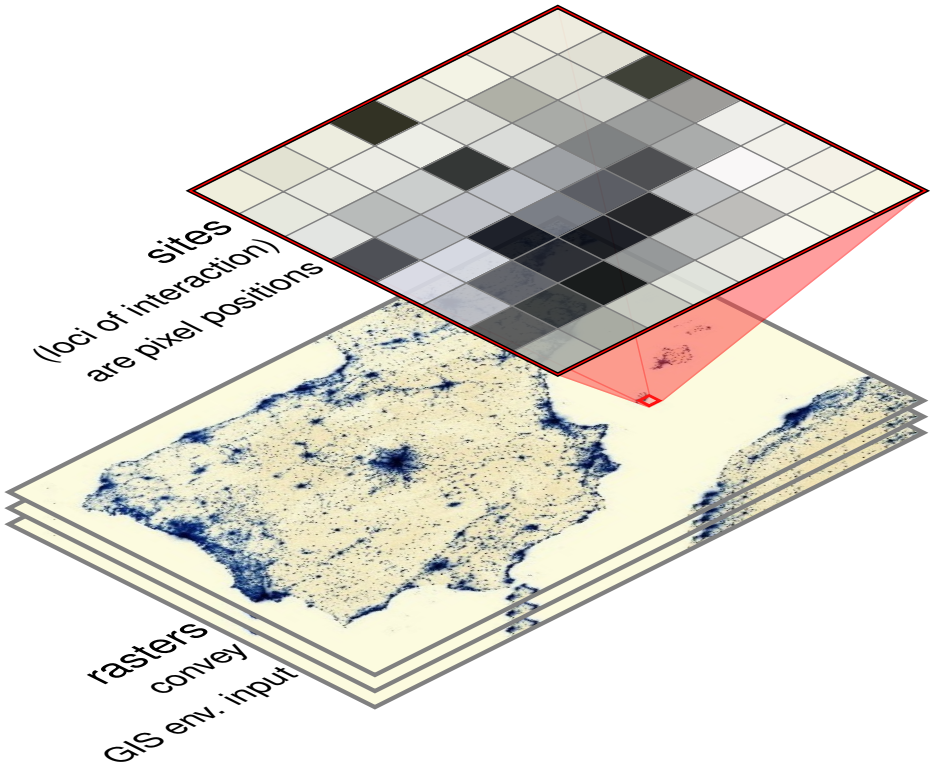
  blaze Eigen MTL4 ...
- CFD

  OpenFOAM® FEniCS ...

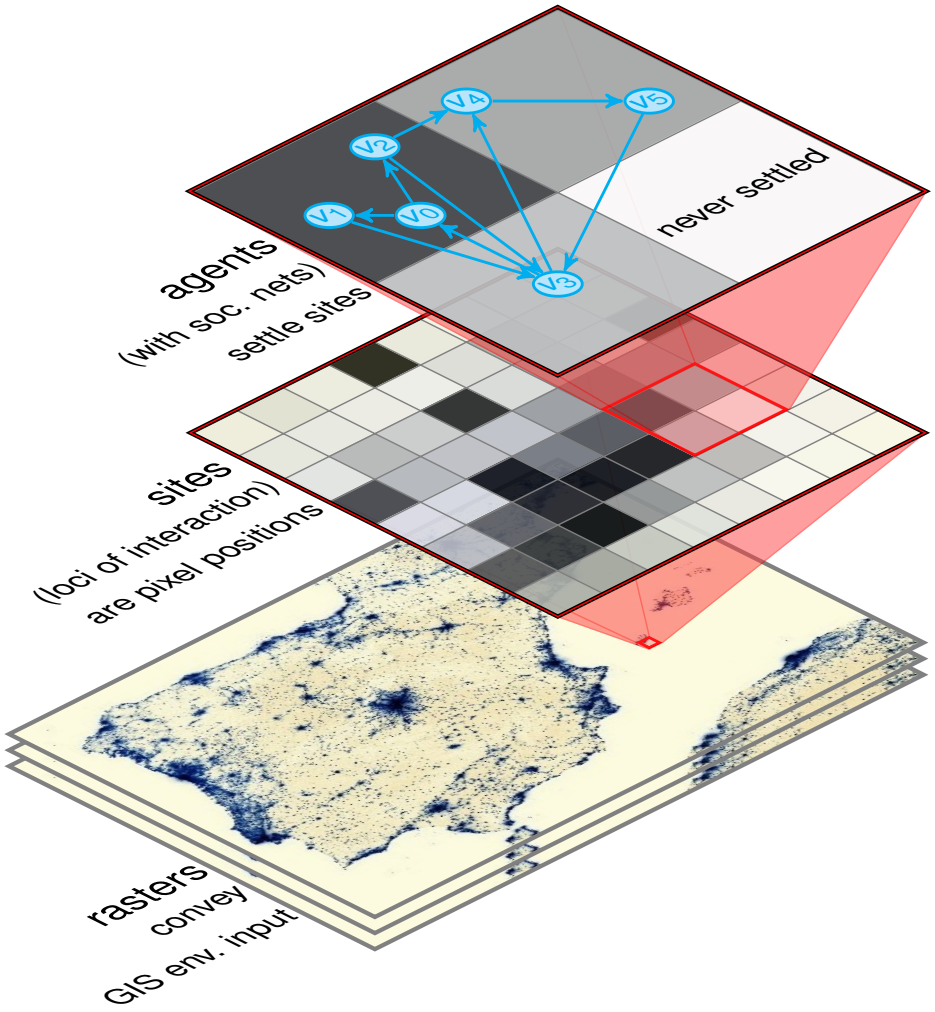
Model : Formalization



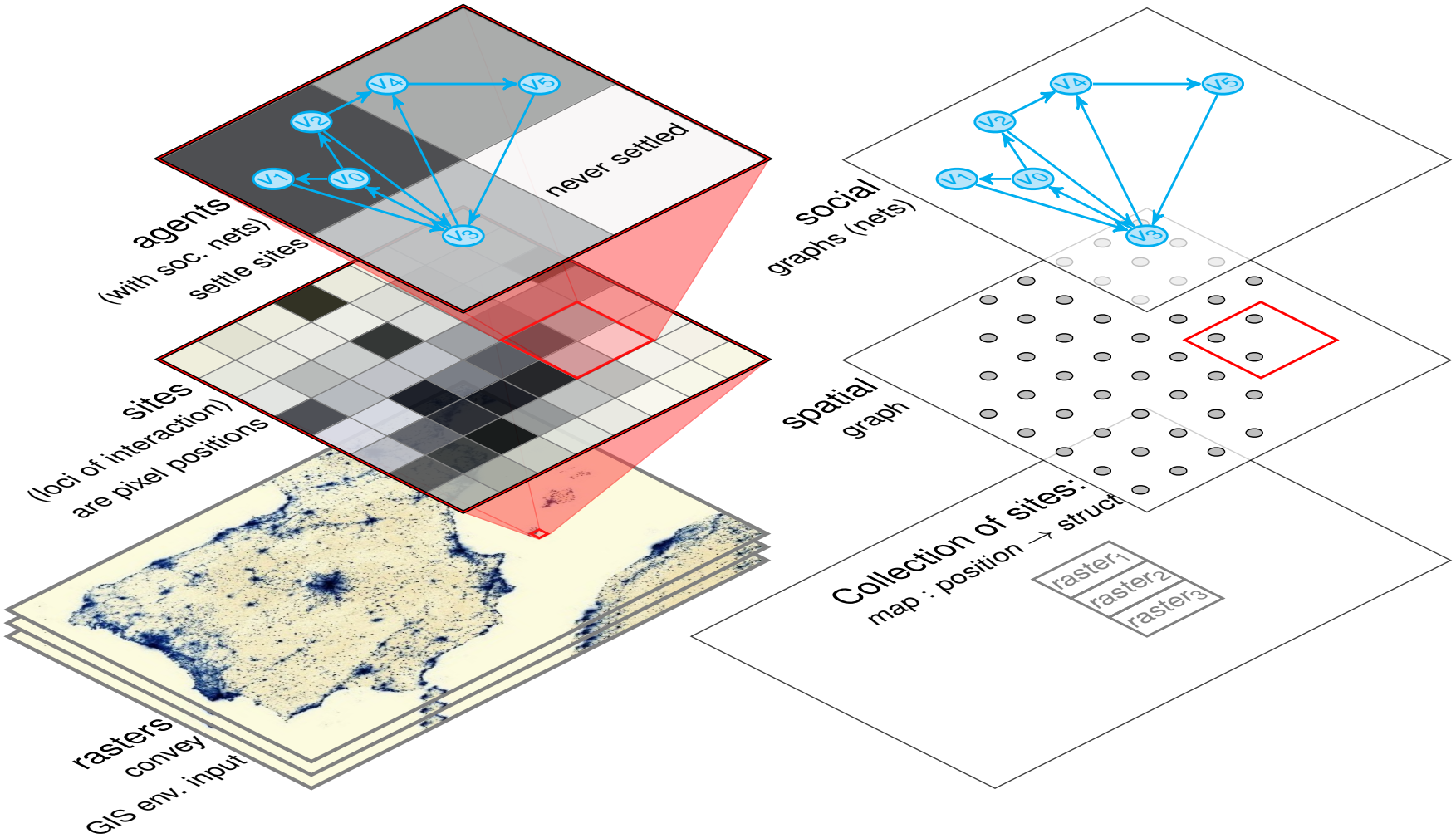
Model : Formalization



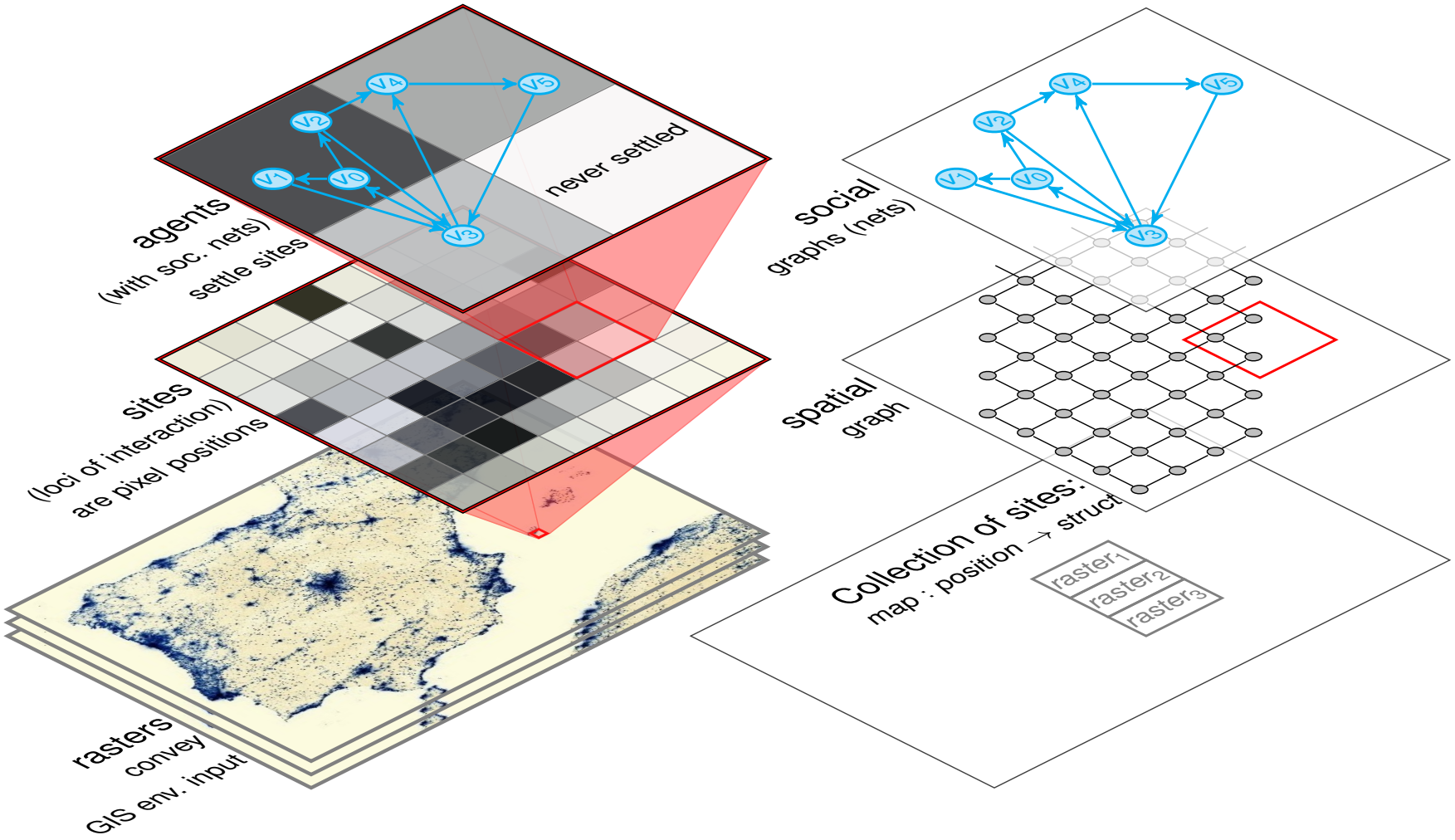
Model : Formalization



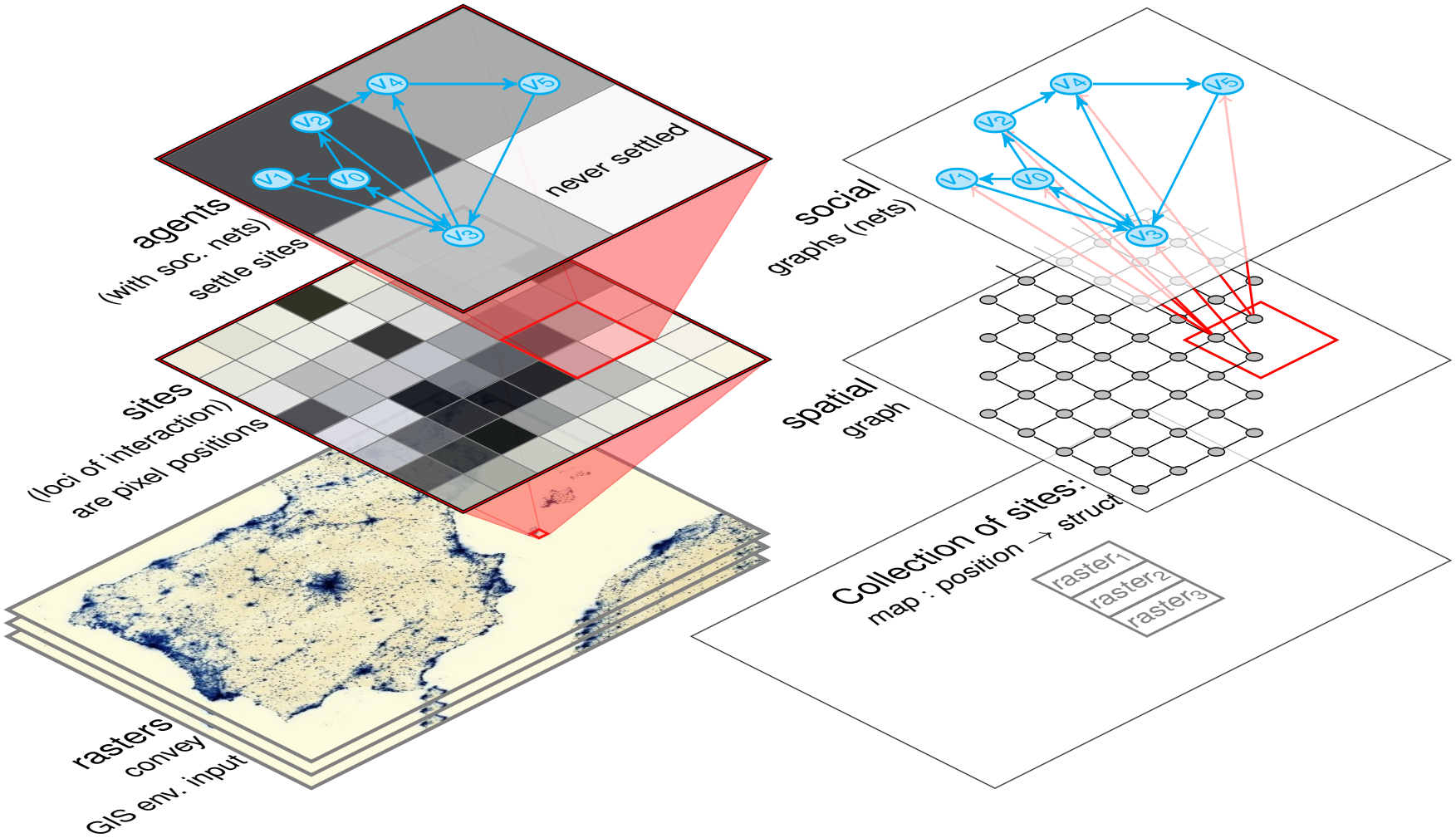
Model : Formalization



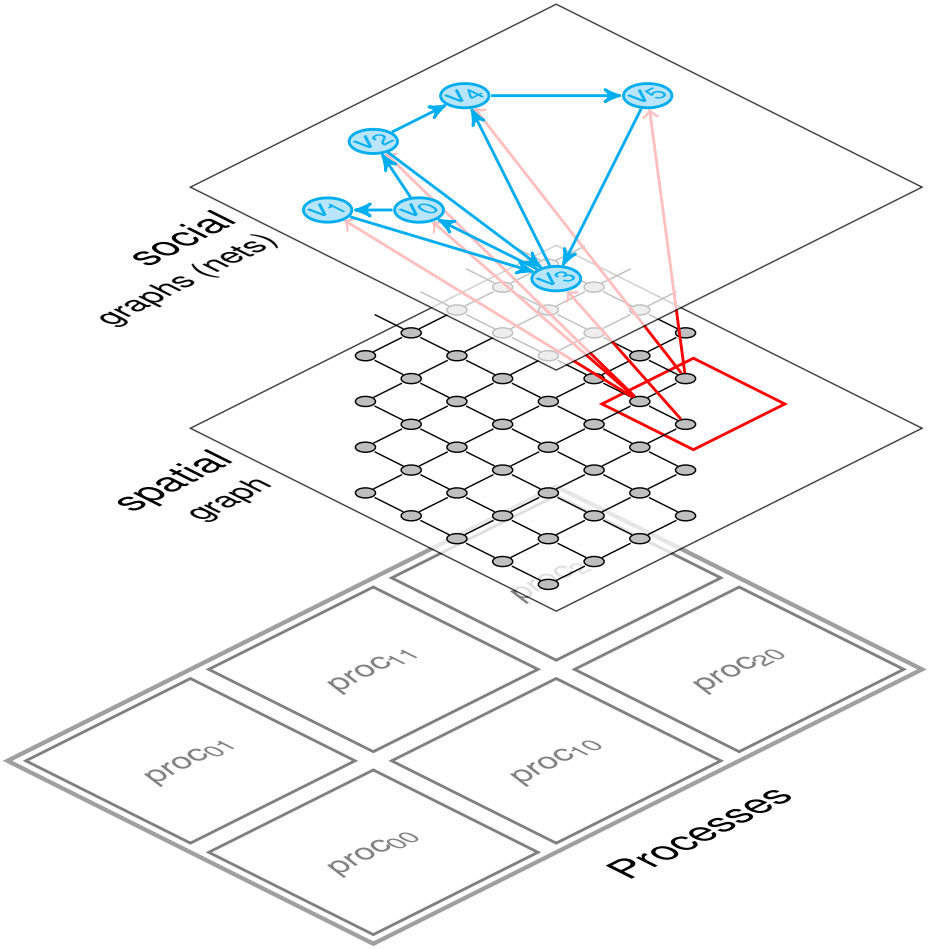
Model : Formalization



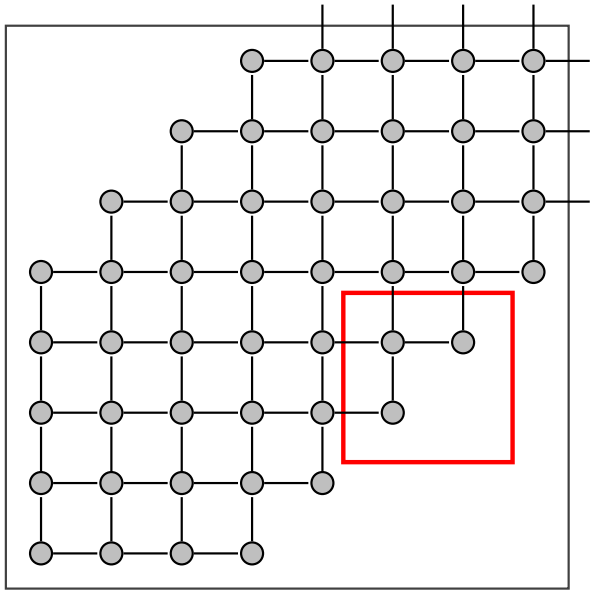
Model : Formalization



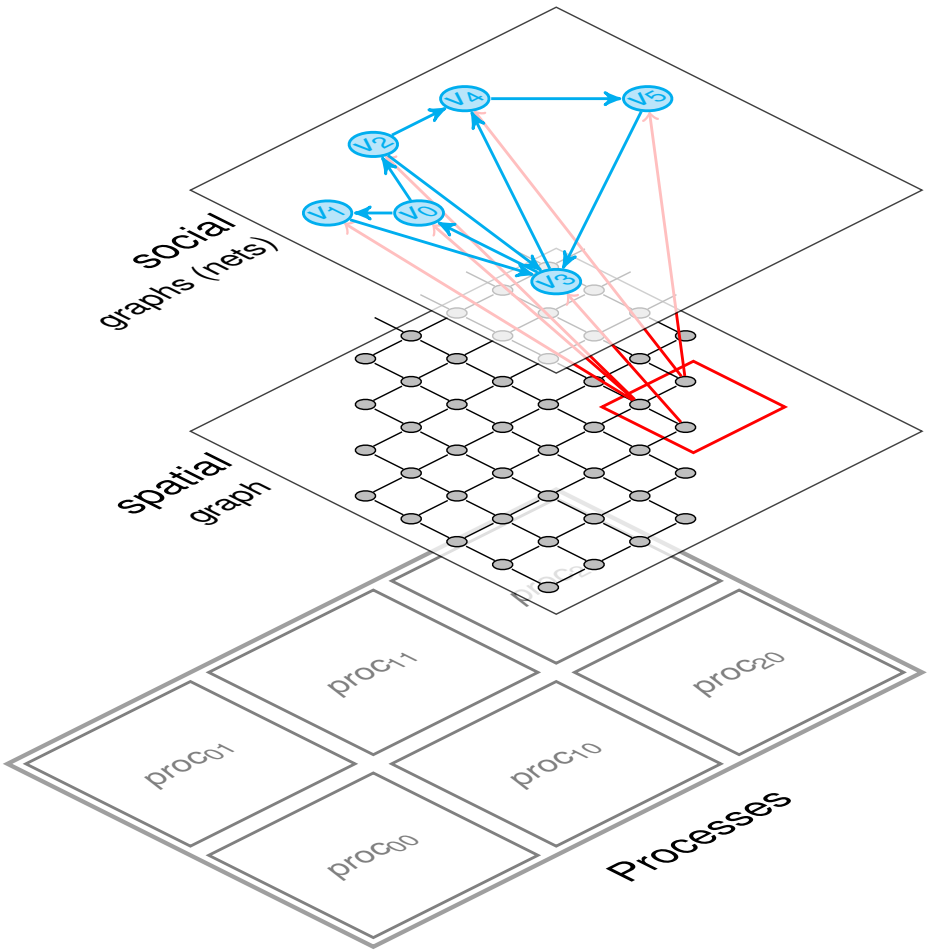
Model : Basic domain decomposition



Computational graph

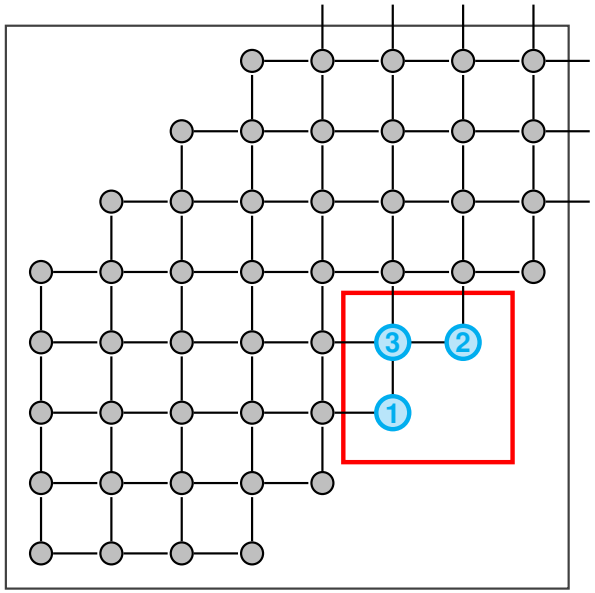


Model : Basic domain decomposition

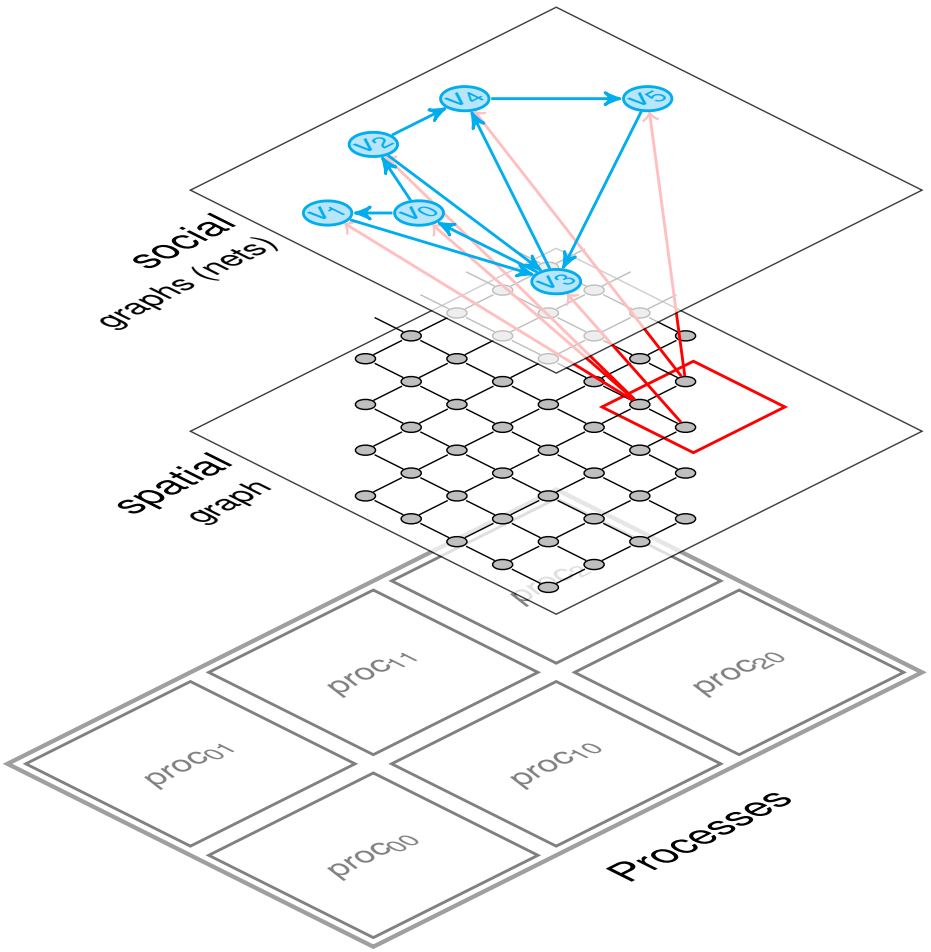


Computational graph

1. vertex weight: $w_i = \#agents_i$

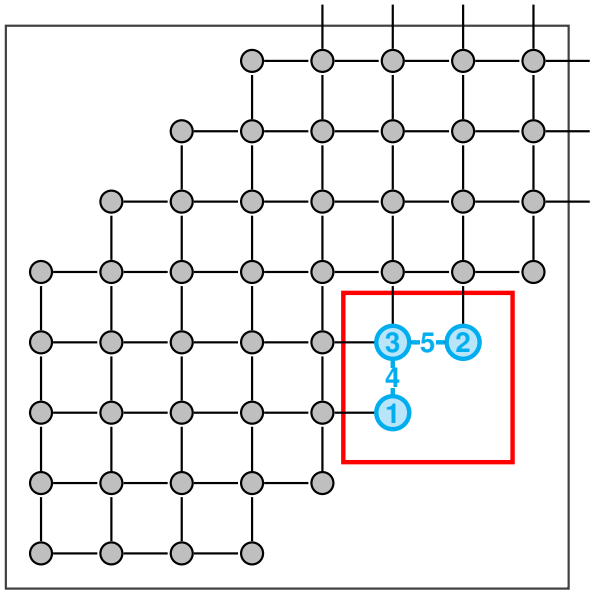


Model : Basic domain decomposition

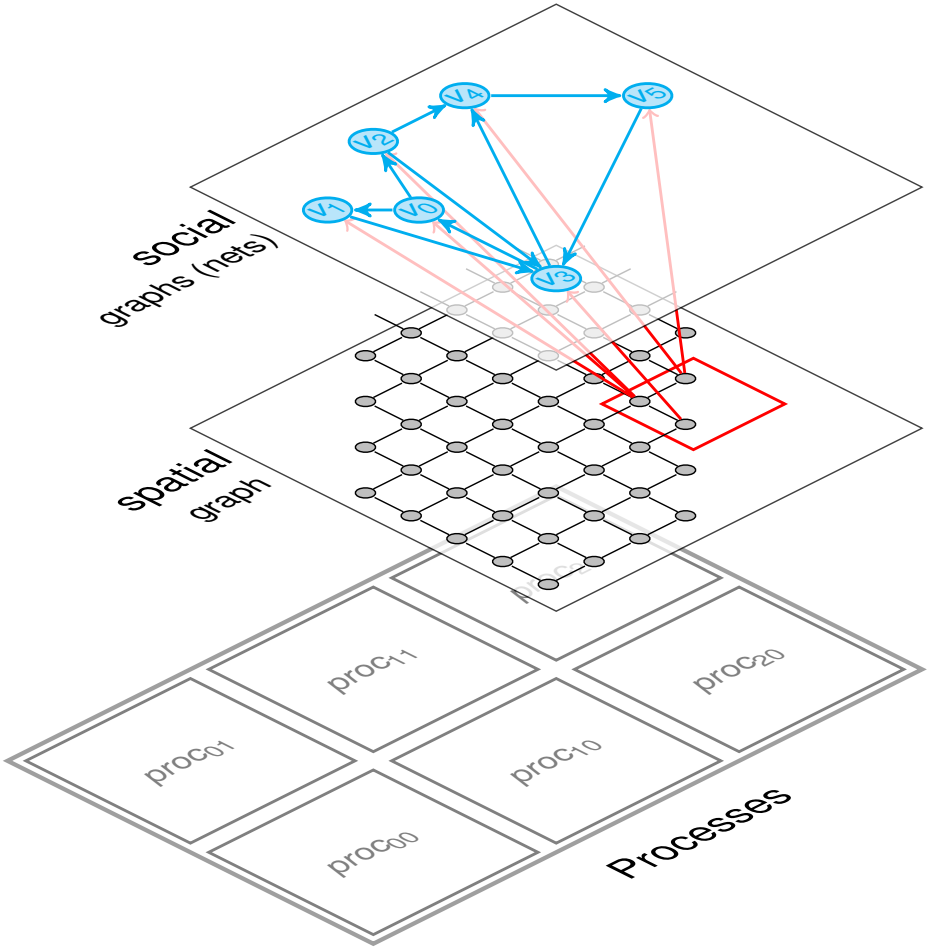


Computational graph

1. vertex weight: $w_i = \#agents_i$
2. SDC-edge weight: $w_{ij} = w_i + w_j$

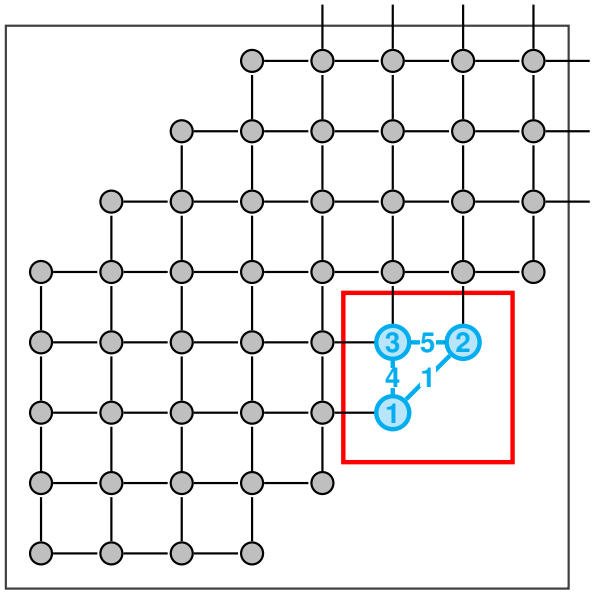


Model : Basic domain decomposition

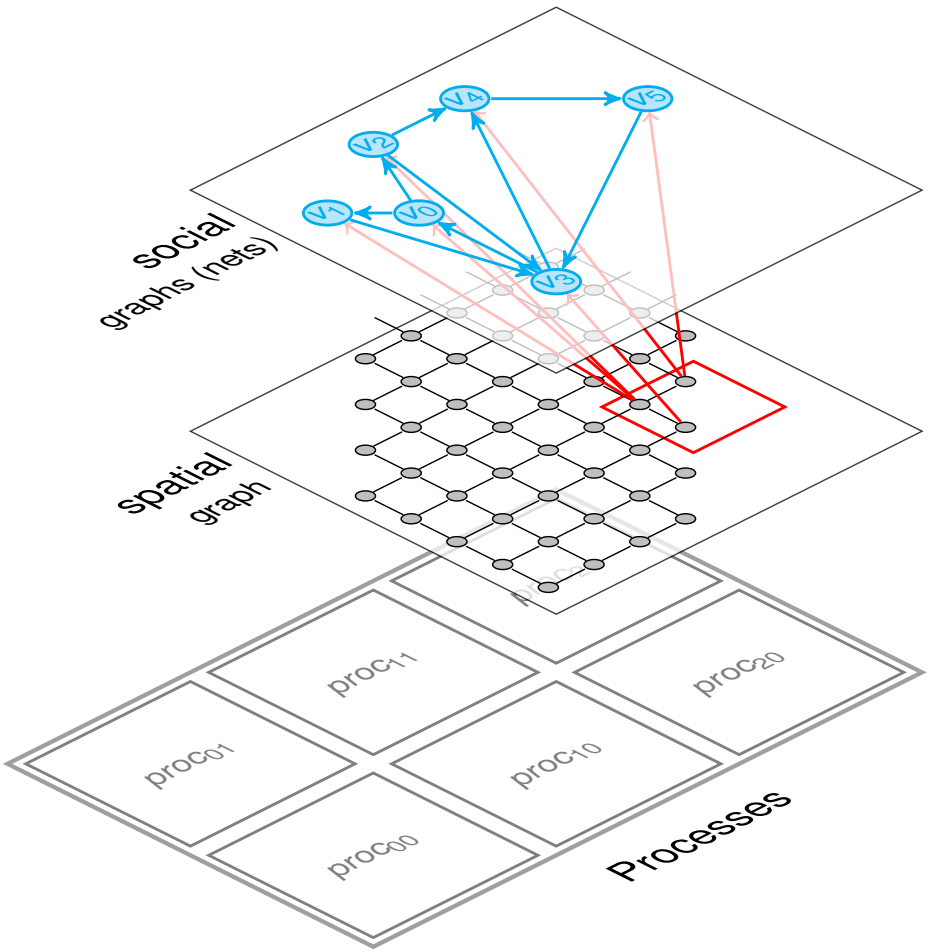


Computational graph

1. vertex weight: $w_i = \#agents_i$
2. SDC-edge weight: $w_{ij} = w_i + w_j$
3. LDC-edge weight: $w_{ij} = \#links_{ij}$

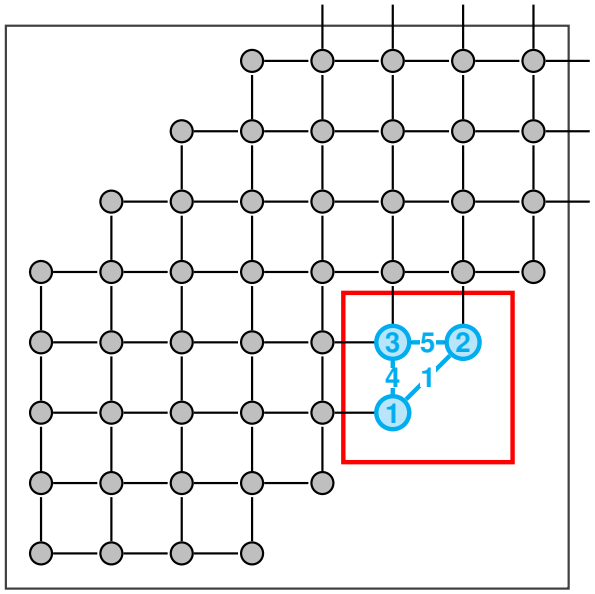


Model : Basic domain decomposition



Computational graph

1. vertex weight: $w_i = \#\text{agents}_i$
2. SDC-edge weight: $w_{ij} = w_i + w_j$
3. LDC-edge weight: $w_{ij} = \#\text{links}_{ij}$



→ **partitioner** (`ParMETIS`, `PT-Scotch`, etc)

Model : Workaround

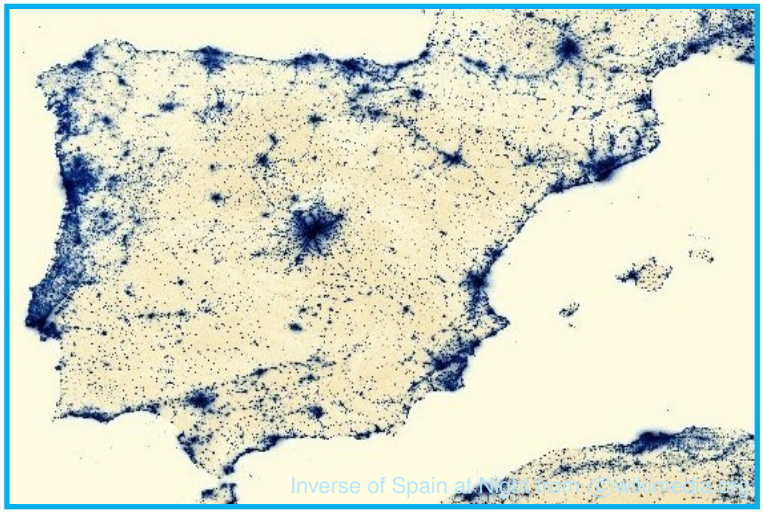
But what if rasters have very high resolution?

Model : Workaround

But what if rasters have very high resolution?

Issue with mapping of sites to processes

- need it for agent move operation

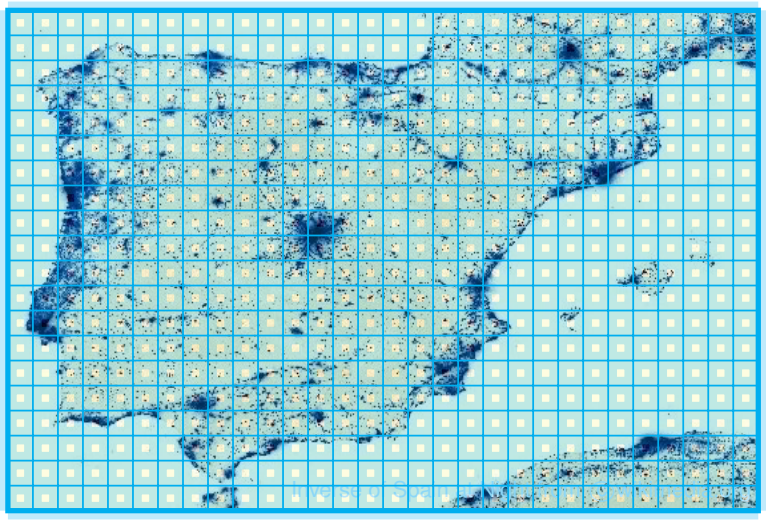


Model : Workaround

But what if rasters have very high resolution?

Issue with mapping of sites to processes

- need it for agent move operation



Computational graph-coarsening

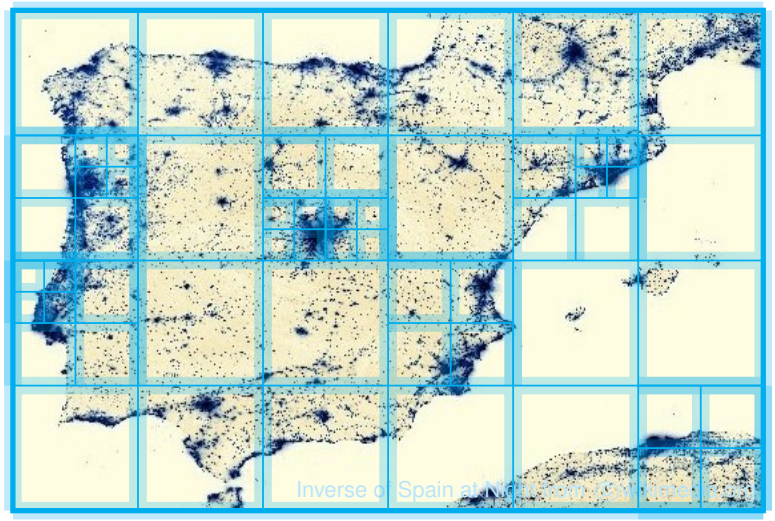
- **uniform mesh**
  - bound min chunk width: $w \geq 2 \cdot halo$
  - disadvantages: disbalanced weights, highly skewed degree distribution

Model : Workaround

But what if rasters have very high resolution?

Issue with mapping of sites to processes

- need it for agent move operation



Computational graph-coarsening

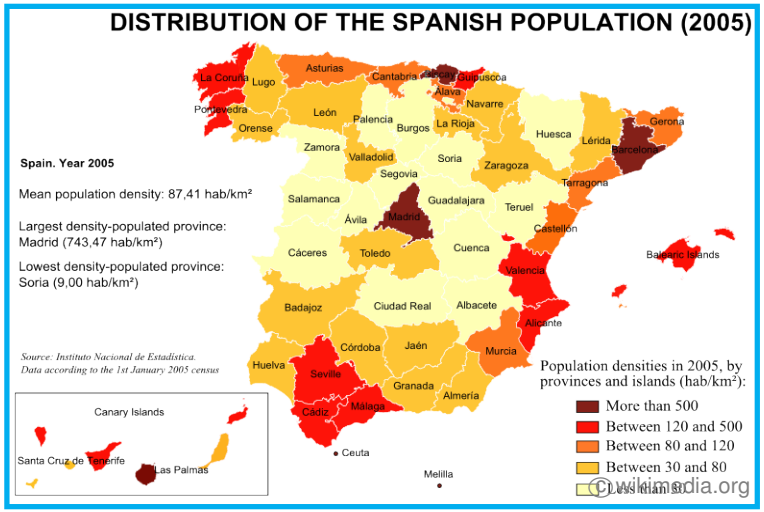
- **uniform mesh**
  - bound min chunk width: $w \geq 2 \cdot halo$
  - disadvantages: disbalanced weights, highly skewed degree distribution
- **multi-scale mesh**
  - quad-tree
    (like in Barnes-Hut algorithm)

Model : Workaround

But what if rasters have very high resolution?

Issue with mapping of sites to processes

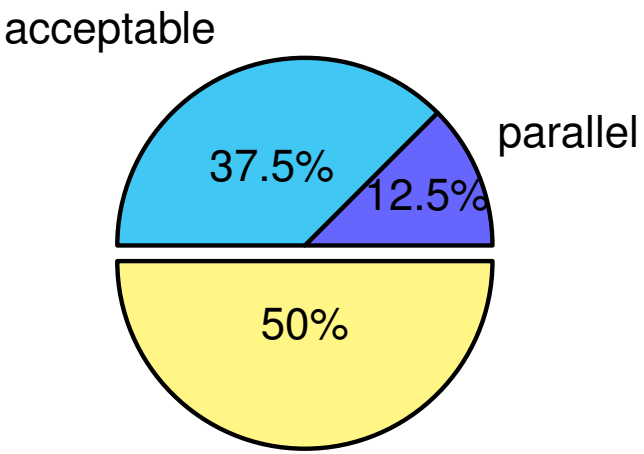
- need it for agent move operation

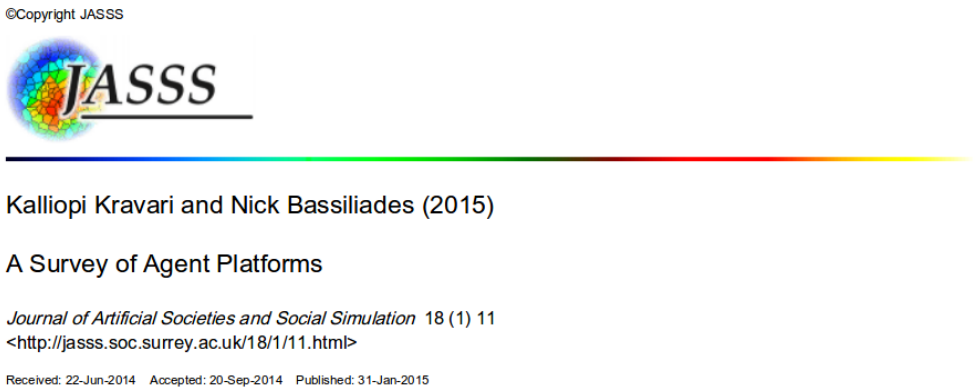


Computational graph-coarsening

- **uniform mesh**
  - bound min chunk width: $w \geq 2 \cdot halo$
  - disadvantages: disbalanced weights, highly skewed degree distribution
- **multi-scale mesh**
  - quad-tree
    (like in Barnes-Hut algorithm)
- **vector GIS information**
  - (local) administrative units $\equiv$ unstr. tree

Software : ABMS Frameworks




©Copyright JASSS

JASSS

Kalliopi Kravari and Nick Bassiliades (2015)

A Survey of Agent Platforms

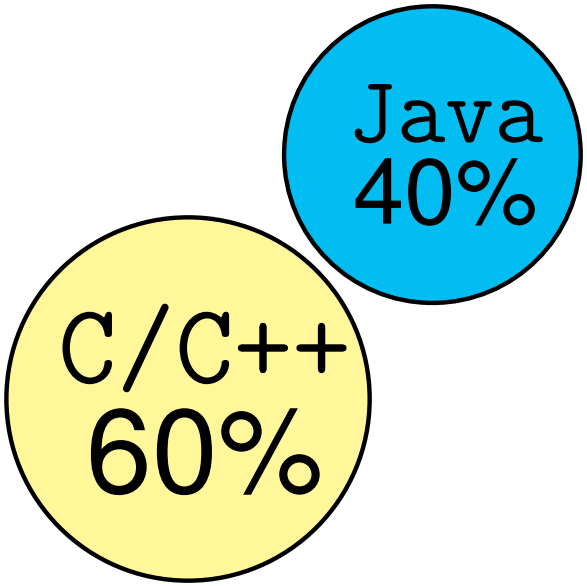
Journal of Artificial Societies and Social Simulation 18 (1) 11
<http://jasss.soc.surrey.ac.uk/18/1/11.html>

Received: 22-Jun-2014 Accepted: 20-Sep-2014 Published: 31-Jan-2015

- 24 popular frameworks
- 3 have parallel facilities (only `RepastHPC` in `C++`)

... rather pessimistic paper
(seems like `RepastHPC` is the only choice)

Software : Distributed ABMS Frameworks for HPC



COMPUTER SCIENCE REVIEW 22 (2016) 27–46

Available online at www.sciencedirect.com
ScienceDirect
journal homepage: www.elsevier.com/locate/cosrev
ELSEVIER

Survey

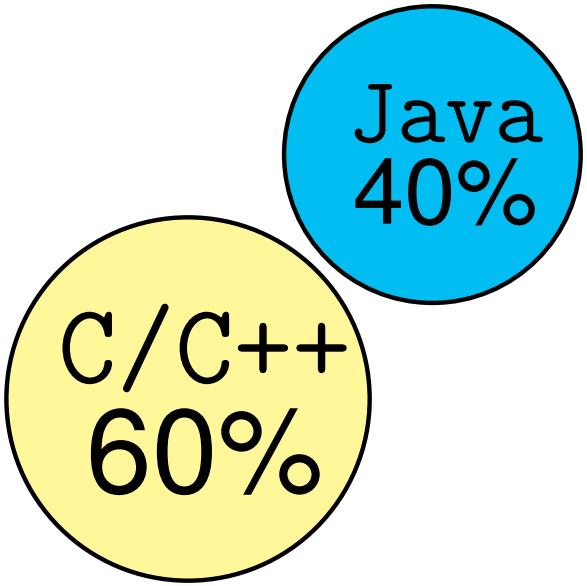
A survey on parallel and distributed multi-agent systems for high performance computing simulations

Alban Rousset, Bénédicte Herrmann, Christophe Lang, Laurent Philippe*

Femto-ST Institute CNRS/UFC – Faculté des Sciences et des Techniques, 16 Route de Gray 25030 Besançon cedex, France

- **10 open source** ABMS frameworks for HPC
- some in `Jave`
  - `D-MASON`, `Jade`, `MACE3J`, `SWAGES`
- **6 out of 10 in `C/C++` (`Charm++`)**
  - recent: `RepastHPC`, `FlameHPC`, `Pandora`
    oldies: `EcoLab`, `PDES-MAS`, `ABM++`

Software : Distributed ABMS Frameworks for HPC



COMPUTER SCIENCE REVIEW 22 (2016) 27–46

Available online at www.sciencedirect.com
ScienceDirect
journal homepage: www.elsevier.com/locate/cosrev
ELSEVIER

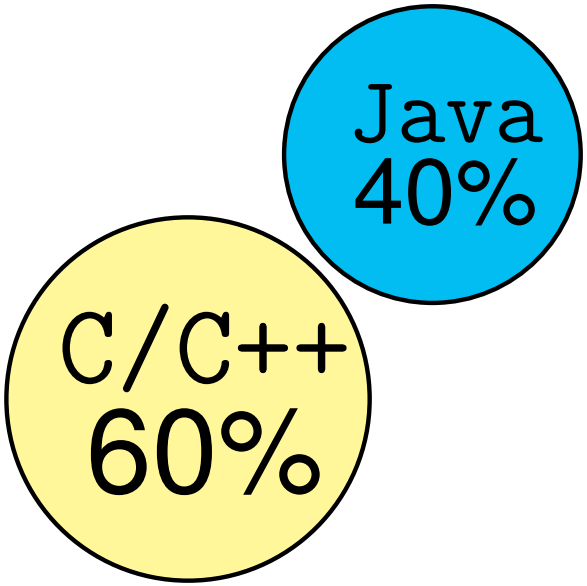
Survey

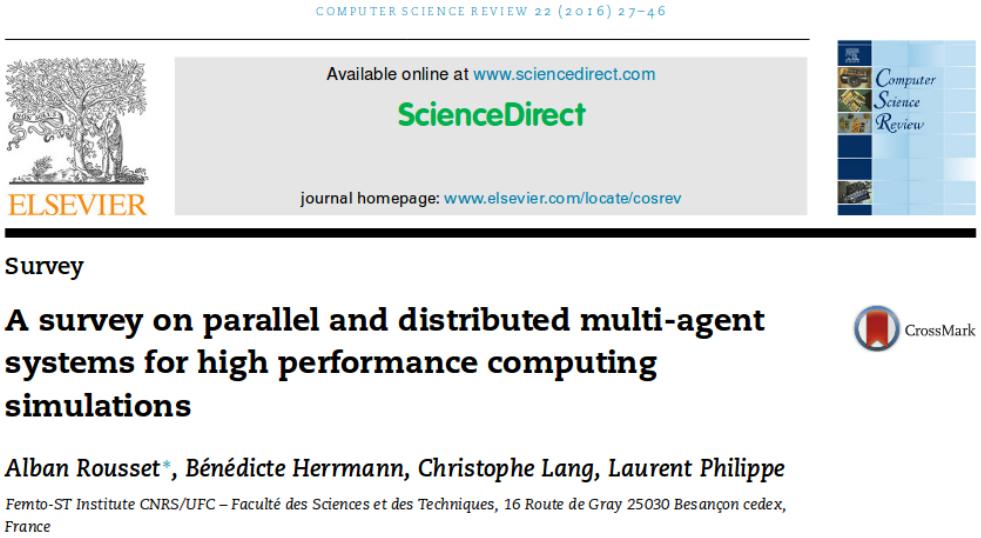
A survey on parallel and distributed multi-agent systems for high performance computing simulations

Alban Rousset, Bénédicte Herrmann, Christophe Lang, Laurent Philippe*
Femto-ST Institute CNRS/UFC – Faculté des Sciences et des Techniques, 16 Route de Gray 25030 Besançon cedex, France

- **10 open source** ABMS frameworks for HPC
- some in `Jave`
  - `D-MASON`, `Jade`, `MACE3J`, `SWAGES`
- **6 out of 10 in `C/C++` (`Charm++`)**
  - recent: `RepastHPC`, `FlameHPC`, `Pandora`
    oldies: `EcoLab`, `PDES-MAS`, `ABM++`
  - beyon the paper scope: `EpiSemdemics`, `FluTE`, ...

Software : Distributed ABMS Frameworks for HPC




COMPUTER SCIENCE REVIEW 22 (2016) 27–46

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cosrev

ELSEVIER

Survey

A survey on parallel and distributed multi-agent systems for high performance computing simulations

Alban Rousset, Bénédicte Herrmann, Christophe Lang, Laurent Philippe*

Femto-ST Institute CNRS/UFC – Faculté des Sciences et des Techniques, 16 Route de Gray 25030 Besançon cedex, France

- **10 open source** ABMS frameworks for HPC
- some in `Jave`
  - `D-MASON`, `Jade`, `MACE3J`, `SWAGES`
- **6 out of 10 in `C/C++` (`Charm++`)**
  - recent: `RepastHPC`, `FlameHPC`, `Pandora`
    oldies: `EcoLab`, `PDES-MAS`, `ABM++`
  - beyon the paper scope: `EpiSemdemics`, `FluTE`, ...

... we need another tool though!

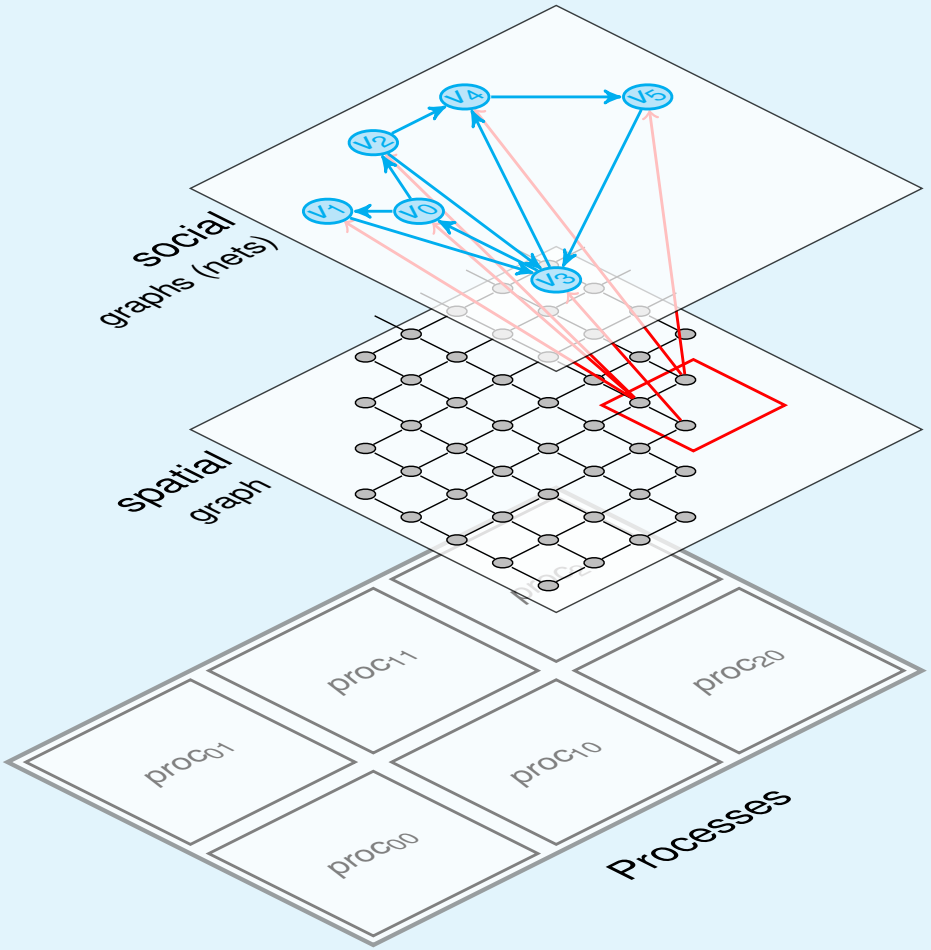
What?
Yet another ABMS framework?

Munch, Edvard. *The Scream* ©wikimedia.org

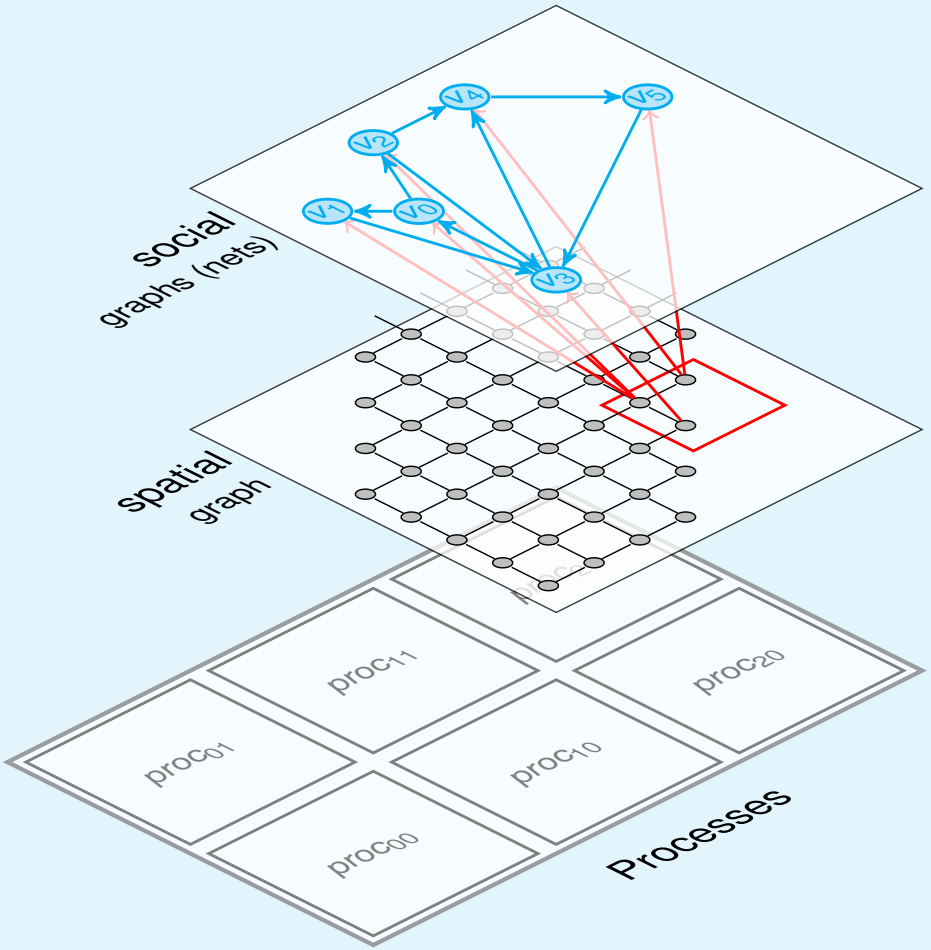
Maybe we can reuse something from existing...

Munch, Edvard. *The Scream* ©wikimedia.org

Software : Building blocks (Primitives)

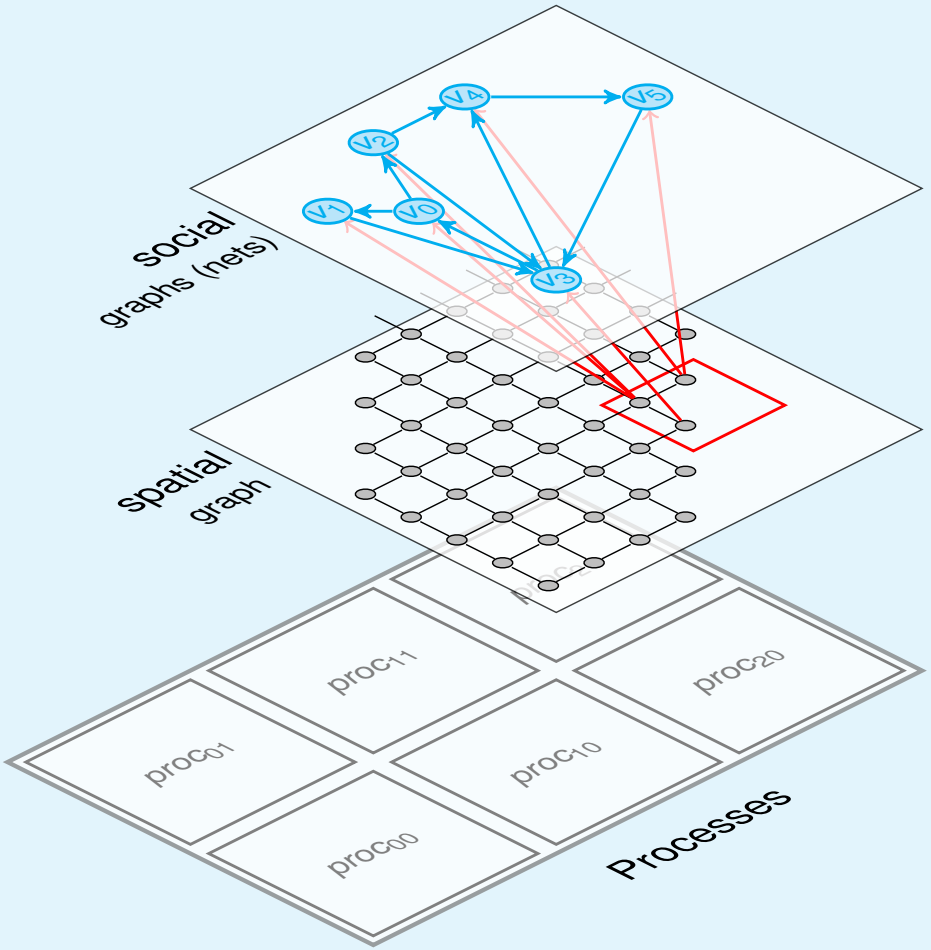


Software : Building blocks (Primitives)



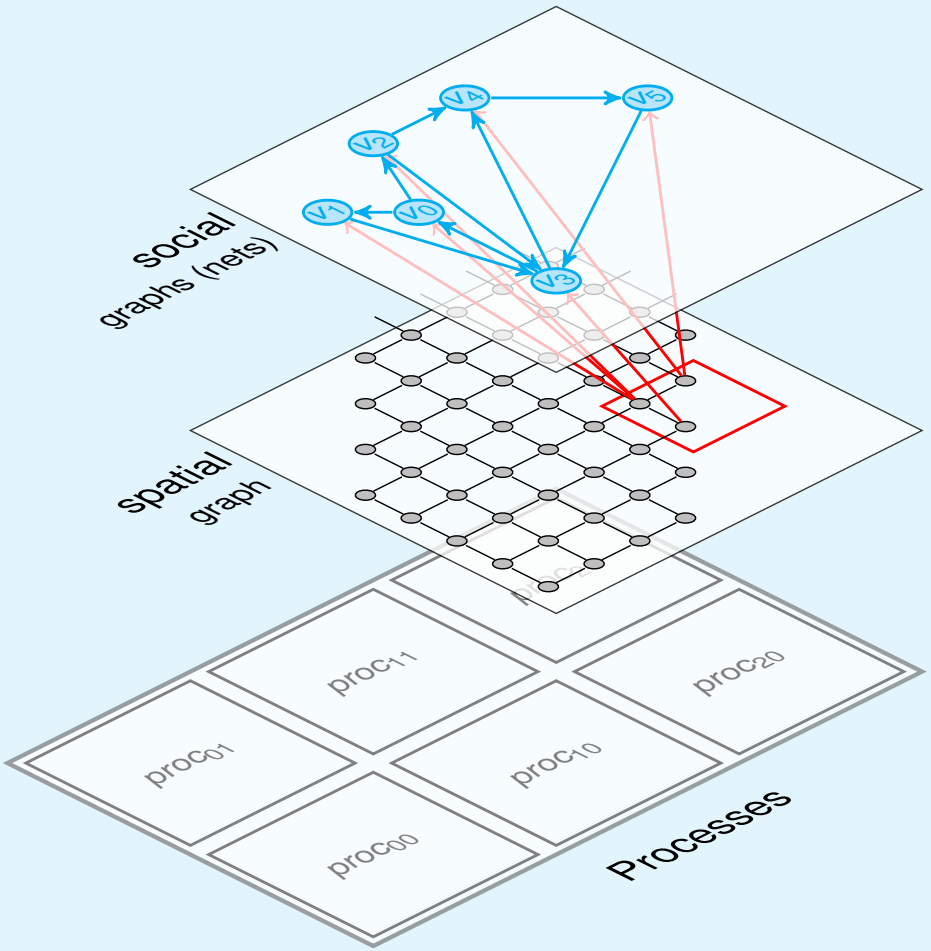
- agents (& synthetic population)
  - type introspection
  - heterogeneous container

Software : Building blocks (Primitives)



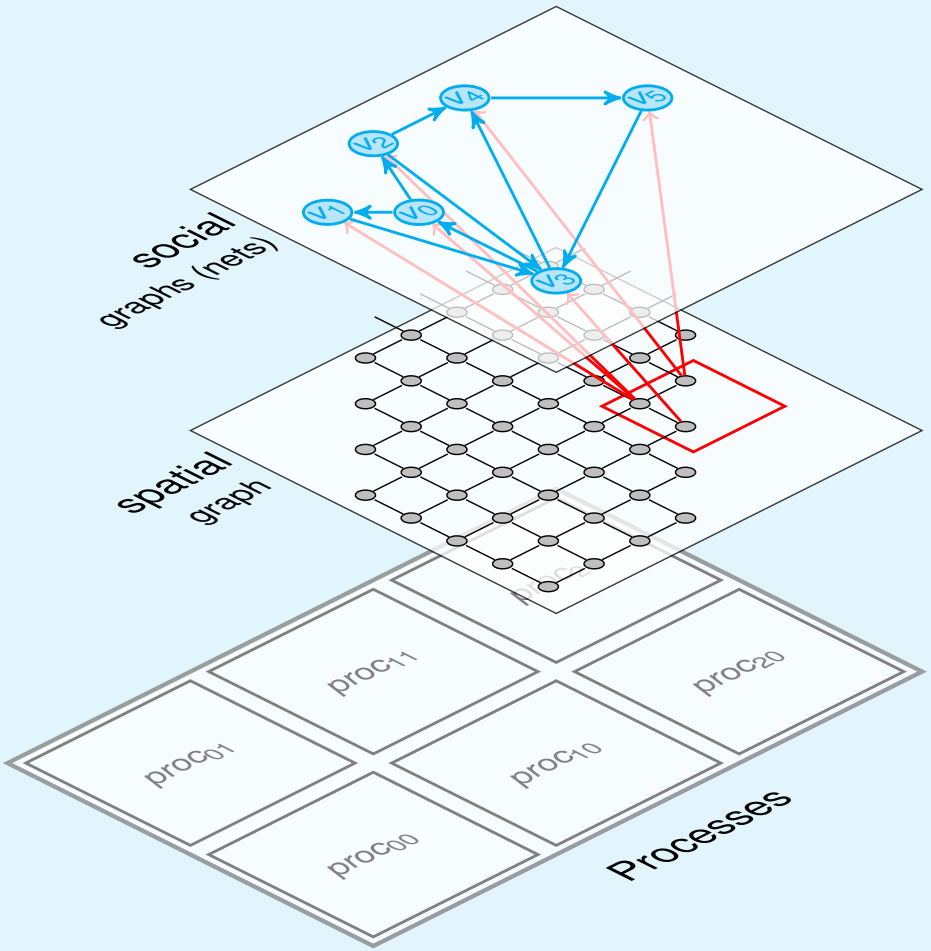
- agents (& synthetic population)
  - type introspection
  - heterogeneous container
- social relationships
  - evolving undirected graphs
  - with and without edge weights

Software : Building blocks (Primitives)



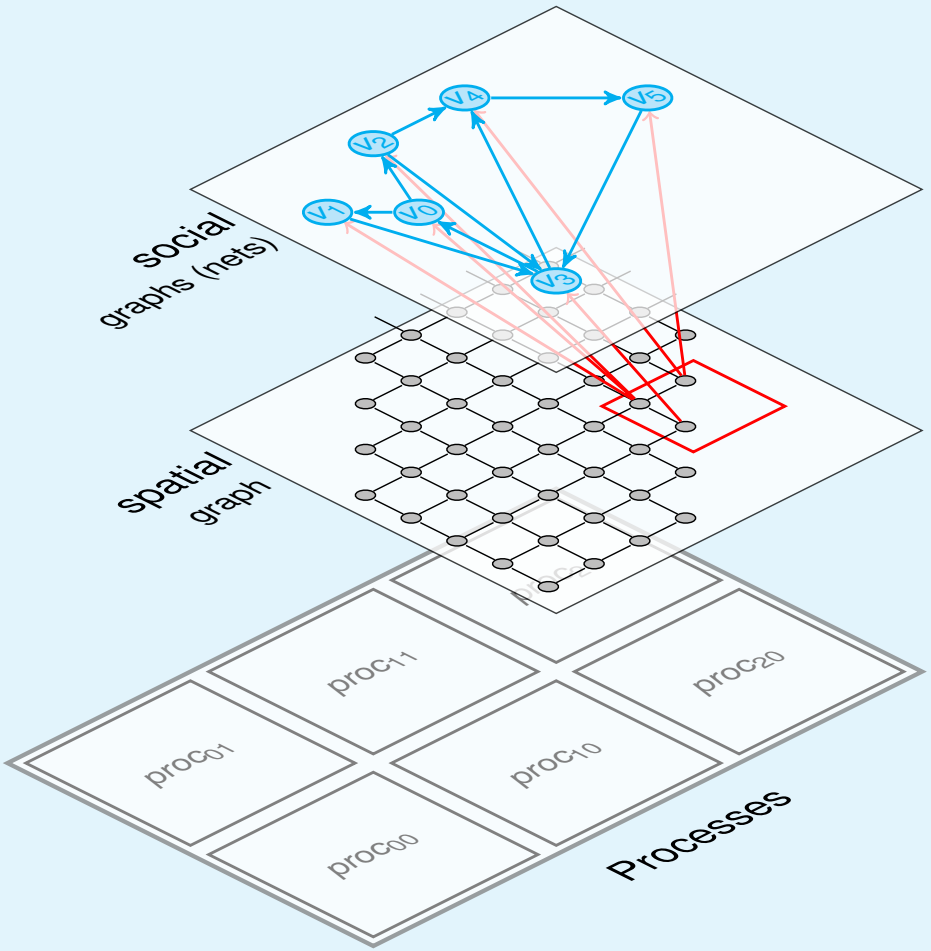
- agents (& synthetic population)
  - type introspection
  - heterogeneous container (fast heterogeneous hash table)
- social relationships
  - evolving undirected graphs
  - with and without edge weights

Software : Building blocks (Primitives)



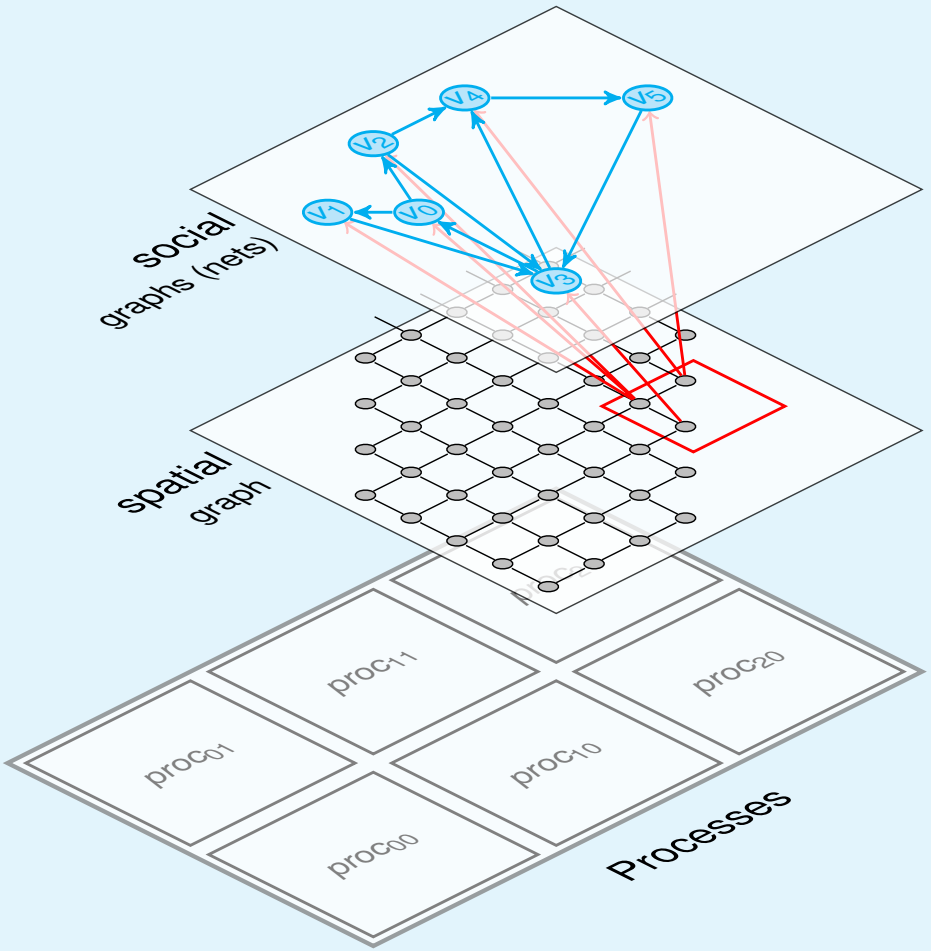
- agents (& synthetic population)
  - type introspection
  - heterogeneous container (fast heterogeneous hash table)
- social relationships
  - evolving undirected graphs
  - with and without edge weights
- spatial information
  - map: position → pixel info

Software : Building blocks (Primitives)



- agents (& synthetic population)
  - type introspection
  - heterogeneous container (fast heterogeneous hash table)
- social relationships
  - evolving undirected graphs
  - with and without edge weights
- spatial information
  - map: position → pixel info
- distribution (domain decomposition)
  - graph coarsening support
    - map: position → coarse piece
  - graph partitioner
  - communication layer

Software : Building blocks (Primitives)



- agents (& synthetic population)
  - type introspection
  - heterogeneous container (fast heterogeneous hash table)
- social relationships
  - evolving undirected graphs
  - with and without edge weights
- spatial information
  - map: position → pixel info
- distribution (domain decomposition)
  - graph coarsening support
    - map: position → coarse piece
  - graph partitioner
  - communication layer
- I/O and checkpointing
  - serialization in HPC compliant format (`HDF5`, `NetCDF`)
  - GIS support

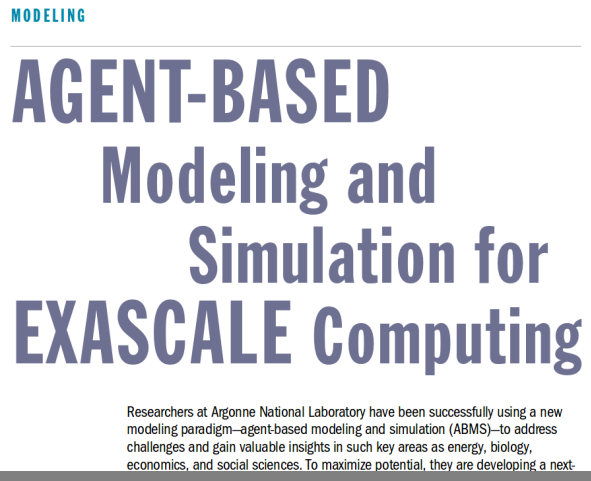
Software : Pandora & RepastHPC

Pandora

- Barcelona Supercomputing Center
  Centro Nacional de Supercomputación
- version: 1
- commits: 630; Jun 17, 2016
- since 2013

repast HPC

- Argonne NATIONAL LABORATORY
- version: 2.2.0
- commits: 249; last Sep 30, 2016
- since 2008


MODELING

AGENT-BASED Modeling and Simulation for EXASCALE Computing

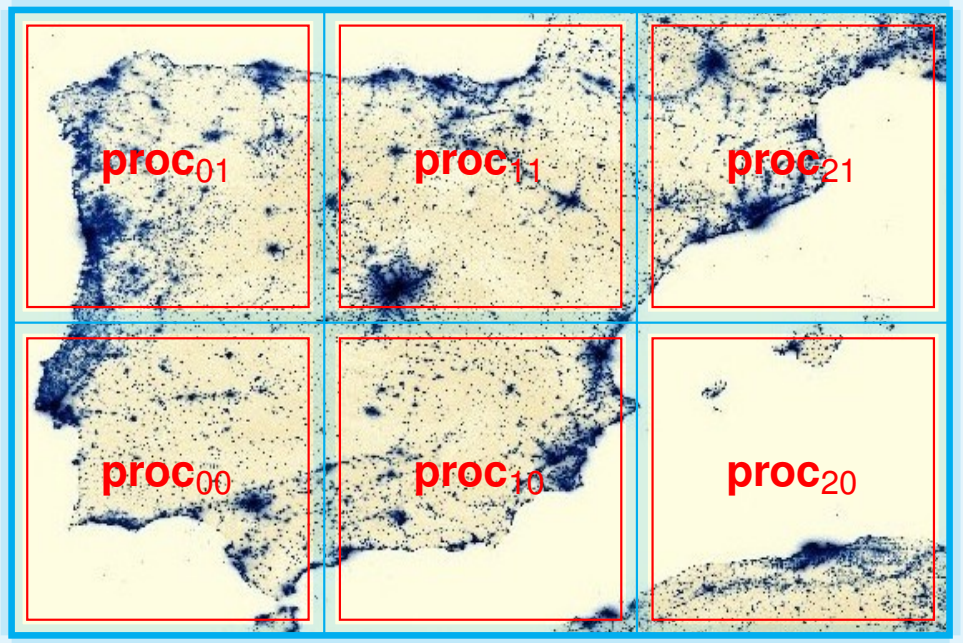
Researchers at Argonne National Laboratory have been successfully using a new modeling paradigm—agent-based modeling and simulation (ABMS)—to address challenges and gain valuable insights in such key areas as energy, biology, economics, and social sciences. To maximize potential, they are developing a next-

Software : Environment Representation

	Pandora	RepastHPC
social relationships representation		
evolving social graphs		+
edge weights		+
multiplicity		+
implementation		native (ptrs)
spatial component representation		
format	`std::map` of dense matrices	grid projector + vector of dense matrices (val layers)
boundary conditions	von Neumann	von Neumann, Moore
GIS support	rasters with `GDAL`	-

Software : Domain Decomposition

“naïve” uniform rectangular mesh

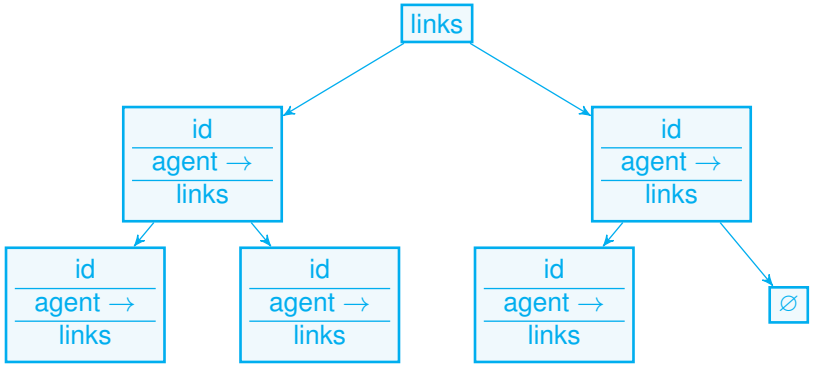


[a]

[a] `RepastHPC` allows to own partitioners written from scratch if one uses network projectors only (e.g., see `chiSIM` project by N.Collier et al.)

Software : Container for Agents

`std::map` (world)



`std::unordered_map` (shared context)

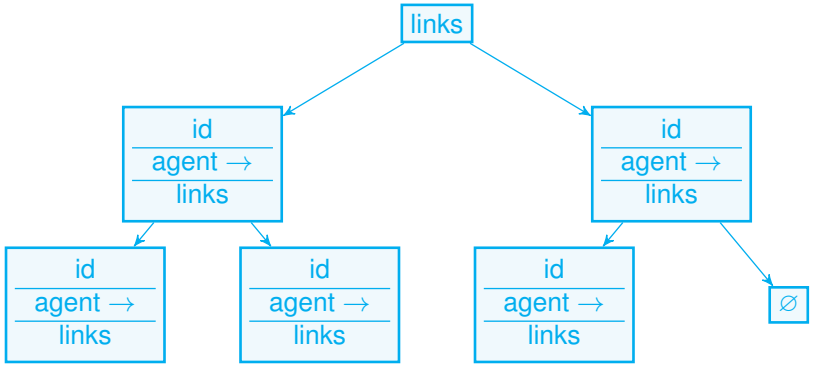
Problems

- `std::map`: $\mathcal{O}(\ln n)$
- `std::unordered_map`?

Can we do better?

Software : Container for Agents

`std::map` (world)



`std::unordered_map` (shared context)

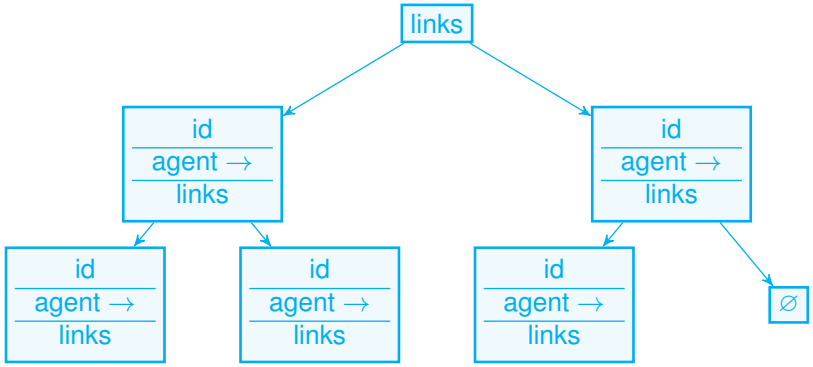
Problems

- `std::map`: $\mathcal{O}(\ln n)$, non-contiguous
- `std::unordered_map`: non-contiguous

Can we do better?

Software : Container for Agents

`std::map` (world)



`std::unordered_map` (shared context)

Problems

- `std::map`: $\mathcal{O}(\ln n)$, non-contiguous
- `std::unordered_map`: non-contiguous

Can we do better?

- Sure! (e.g., `google::dense_hash_map`)
  see, e.g., `tessil` benchmark of major hash maps

Software : Agent Types and Introspection

Compile-time introspection substitutes

`Pandora`

- code generation (+ some boilerplates)
  - error-prone pseudo-parsing with `Python`

`RepastHPC`

- boilerplates
  - Package pattern

Do some work at run-time

Software : Agent Types and Introspection

Compile-time introspection substitutes

`Pandora`

- code generation (+ some boilerplates)
  - error-prone pseudo-parsing with `Python`

`RepastHPC`

- boilerplates
  - Package pattern

Do some work at run-time

Why? Can we do better?

- compile-time introspection was a headache in `C++` up to `C++14` standard

Software : Agent Types and Introspection

Compile-time introspection substitutes

`Pandora`

- code generation (+ some boilerplates)
  - error-prone pseudo-parsing with `Python`

`RepastHPC`

- boilerplates
  - Package pattern

Do some work at run-time

Why? Can we do better?

- compile-time introspection was a headache in `C++` up to `C++14` standard
  - `Boost::HANA` (by Louis Dionne) → improved `Boost::Fusion`
  - Apache `Thrift` (and its Facebook brench)
- since `C++17` we can convert POD-`struct`→`std::tuple` non-intrusively without macro and external tooling (A. Polukhin. *CppCon*, 2016)
- since `C++20` reflection might be part of standard

No more destruction!

Brullov, Karl. *The Last Day of Pompeii* ©wikimedia.org

Let's construct...

What do we have?

- `RepastHPC` **is a good** tool

Let's construct...

What do we have?

- `RepastHPC` **is a good** tool for years 2008-2013
- ... though we **can do a way better** with modern `C++` (since `C++14`) and third-parties
  - compile-time introspection, lambdas + `std::function`, `std::tuple`, ...

Let's construct...

What do we have?

- `RepastHPC` **is a good** tool for years 2008-2013
- ... though we **can do a way better** with modern `C++` (since `C++14`) and third-parties
  - compile-time introspection, lambdas + `std::function`, `std::tuple`, ...
- recently we started to work on a **new framework**
  - prepare "state-of-the-art" HPC compliant primitives for GSS users
  - strive both user-friendliness (EDSL like API) and performance

Let's construct...

What do we have?

- `RepastHPC` **is a good** tool for years 2008-2013
- ... though we **can do a way better** with modern `C++` (since `C++14`) and third-parties
  - compile-time introspection, lambdas + `std::function`, `std::tuple`, ...
- recently we started to work on a **new framework**
  - prepare "state-of-the-art" HPC compliant primitives for GSS users
  - strive both user-friendliness (EDSL like API) and performance
- advances in this development... modest so far
  - **prototype** with run-time heterogeneity mechanicms and introspection
  - use `SNAP` graph library for dealing with social relationships
  - graph coarsening with uniform meshes
  - HDF5 I/O format and its implementation (Rafal Lichwala, PSNC)

Let's construct...

What do we have?

- `RepastHPC` **is a good** tool for years 2008-2013
- ... though we **can do a way better** with modern `C++` (since `C++14`) and third-parties
  - compile-time introspection, lambdas + `std::function`, `std::tuple`, ...
- recently we started to work on a **new framework**
  - prepare "state-of-the-art" HPC compliant primitives for GSS users
  - strive both user-friendliness (EDSL like API) and performance
- advances in this development... modest so far
  - **prototype** with run-time heterogeneity mechanicms and introspection
  - use `SNAP` graph library for dealing with social relationships
  - graph coarsening with uniform meshes
  - HDF5 I/O format and its implementation (Rafal Lichwala, PSNC)

What do we plan and work on?

- shift to extensive use of **compile-time**: templates, introspection, tuples, CRT, etc
- **more third parties**: hash tables, graph libraries
- graph coarsening with **multi-scale meshes**

Thank you for

your patience!

Don’t blame me only! Don’t cheer me only!

Michael Gienger

Ralf Schneider

Bastian Koller