

P&S Heterogeneous Systems

Accelerating Agent-Based Simulations with BioDynaMo

Lukas Breitwieser

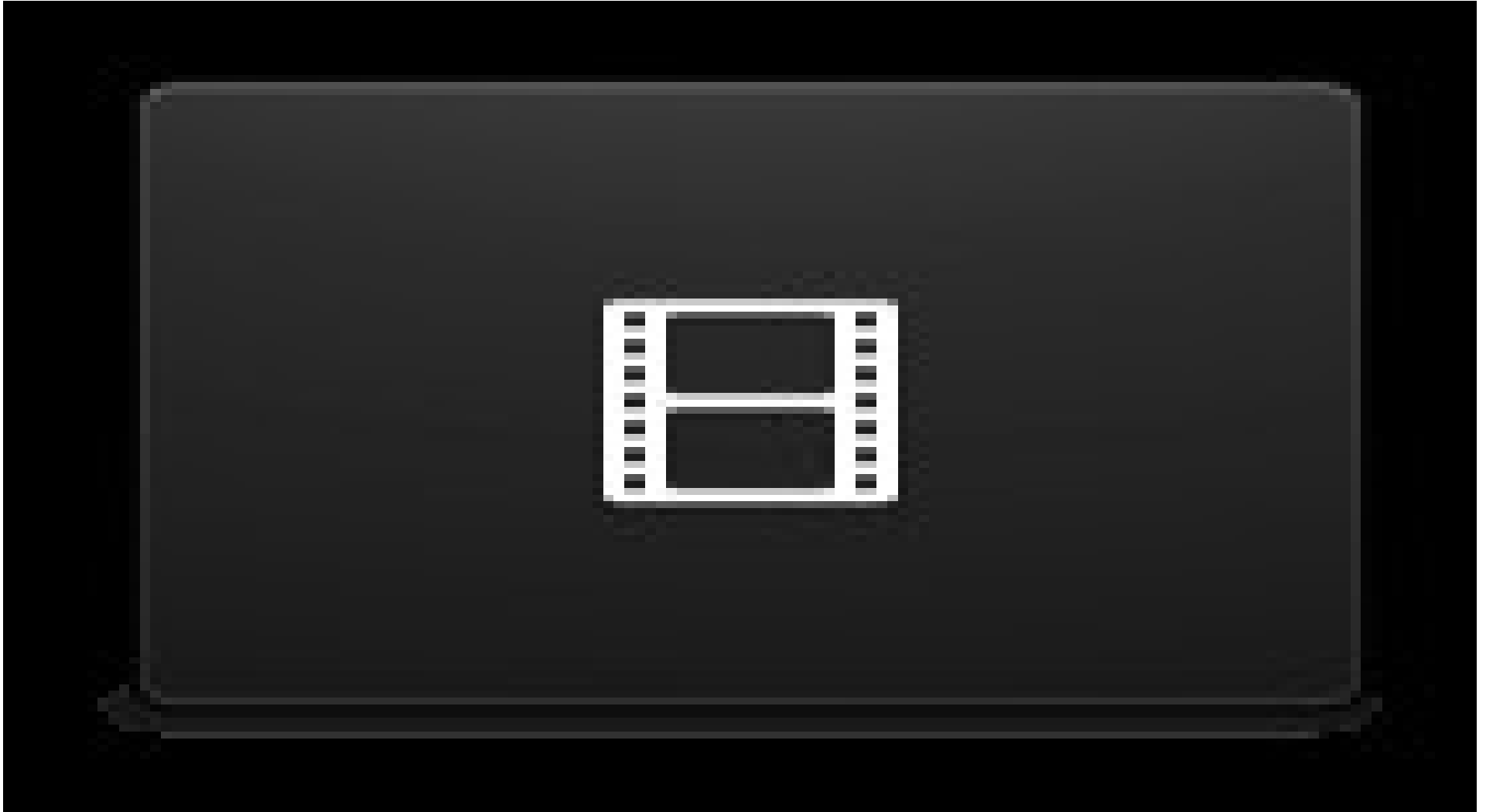
ETH Zürich

Spring 2022

11 July 2022

Introduction to Agent-Based Simulation

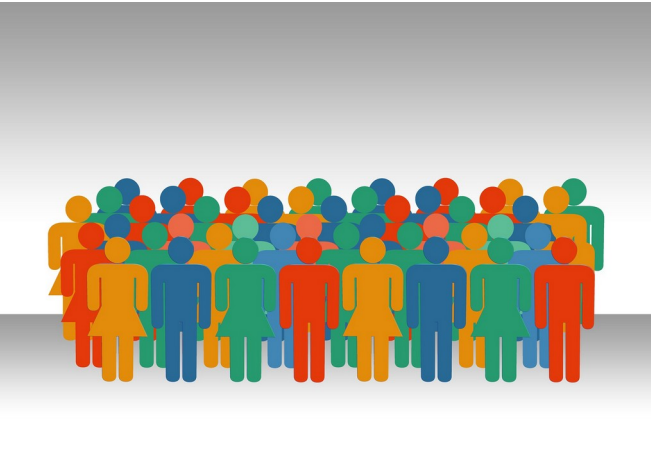
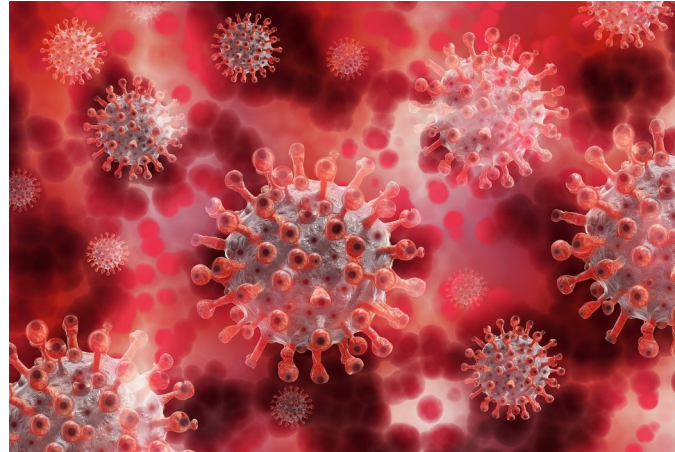
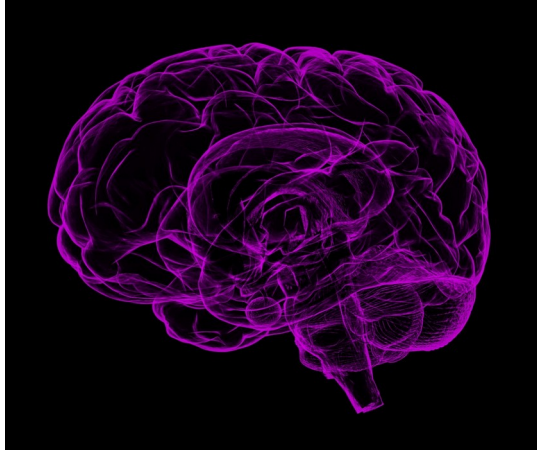
Modeling complex systems



What is agent-based simulation?

- Also called individual-based modeling
- Bottom-up approach
 - Modeling the trees not the forest
- Characteristics
 - Local interaction
 - Emergent behavior

Agent-based simulation is very versatile



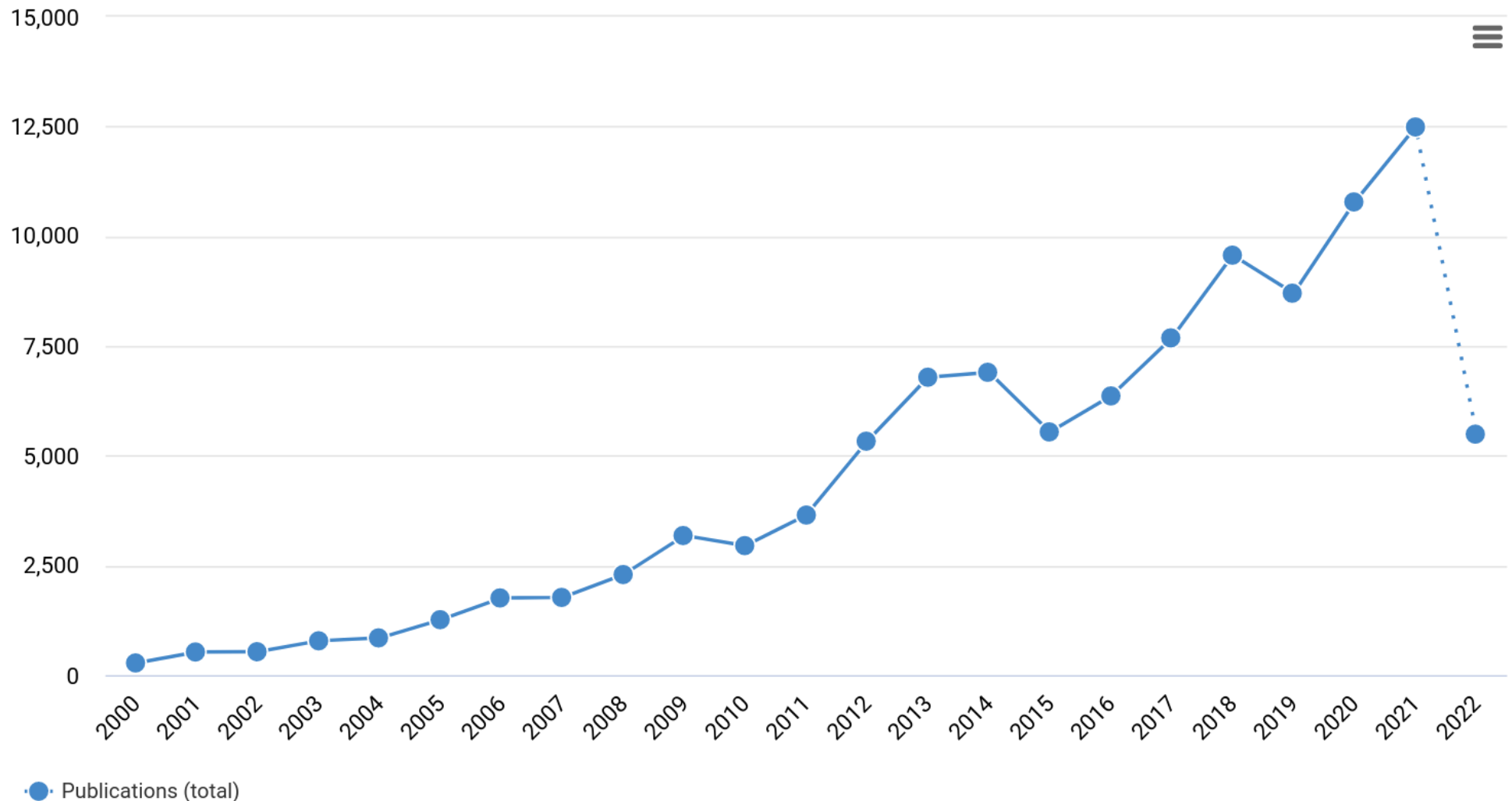
Agent-based simulation is very versatile (cont.)

Table 1: A sample of recent agent-based applications

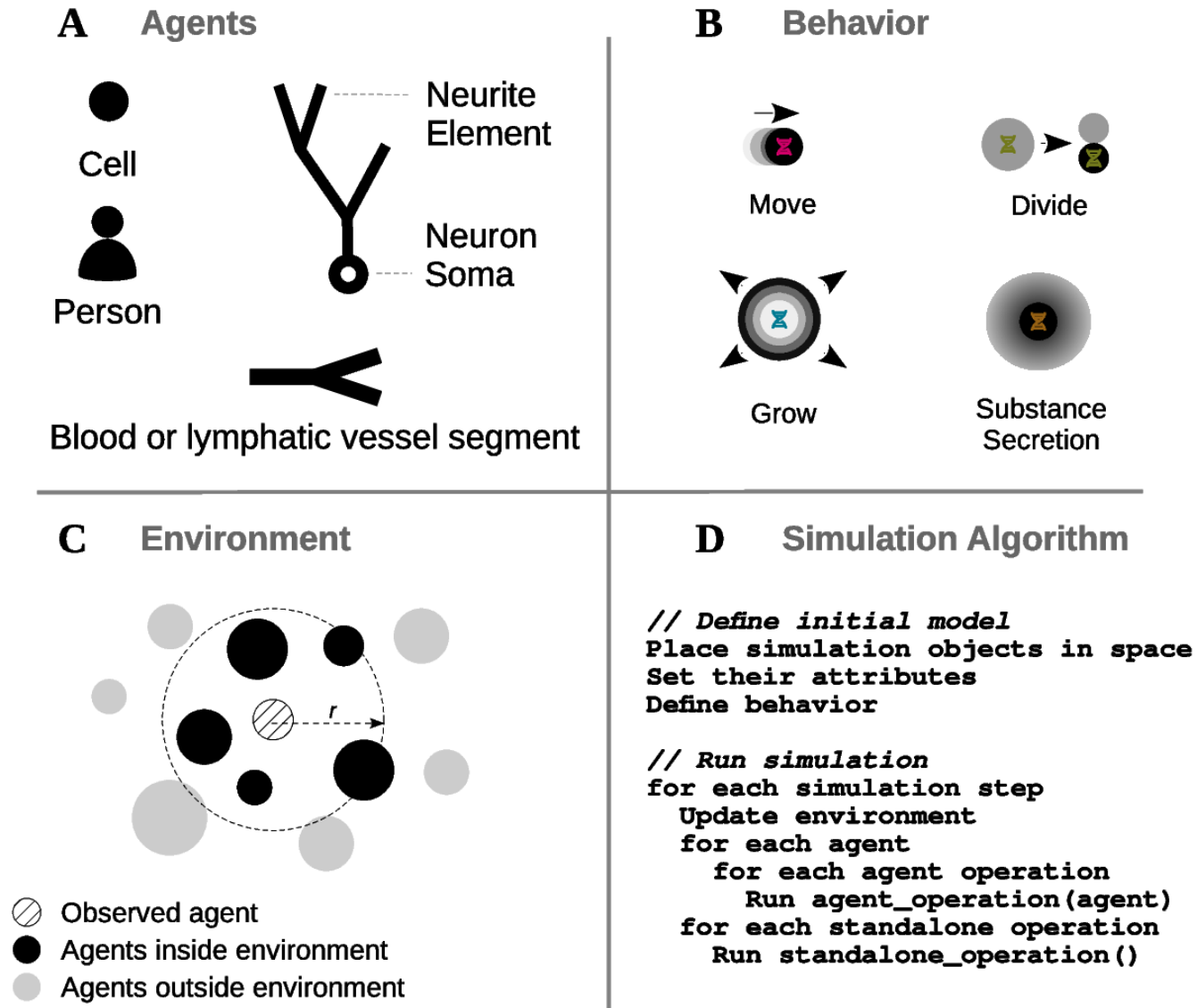
Application Area:	Agent-based Model Focus:
Agriculture	A spatial individual-based model prototype for assessing potential pesticide exposure of farm-workers conducting small-scale agricultural production (Leyk et al. 2009)
Air Traffic Control	Air traffic control to analyze control policies and performance of an air traffic management facility (Conway 2006)
Anthropology	Prehistoric settlement patterns and political consolidation in the Lake Titicaca basin of Peru and Bolivia (Griffin and Stanish 2007)
Biomedical Research	<i>The Basic Immune Simulator</i> , to study the interactions between innate and adaptive immunity (Folcik et al. 2007)
Crime Analysis	A realistic virtual urban environment, populated with virtual burglar agents (Malleson 2010)
Ecology	Investigate the trade-off between road avoidance and salt pool spatial memory in the movement behavior of moose (Grosman et al. 2011) Predator-prey relationships between transient killer whales and other marine mammals (Mock and Testa 2007)
Energy Analysis	A building occupant network energy consumption decision-making model (Chen et al. 2011) Application for the Smart Grid (Jackson 2010) Energy investment decision making (Tobias 2008) Oil refinery supply chain (Van Dam et al. 2008)
Epidemiology	Pandemic disease model accounting for individual behavior and demographics (Aleman et al. 2010) Global-scale agent model of disease transmission (Parker and Epstein 2011)
Evacuation	Tsunami evacuation using a modified form of Helbing's social-force model applied to agents (Puckett 2009)
Market Analysis	Consumer marketing model developed in collaboration with a Fortune 50 firm (North et al. 2009) Consumer airline market share (Kuhn et al. 2010) Simulation that models the possibilities for a future market in sub-orbital space tourism (Charania et al. 2006)
Social Networks	Model of email-based social networks, in which individuals establish, maintain and allow atrophy of links through contact-lists and emails (Menges et al. 2008)

Rising Number of Publications in this field

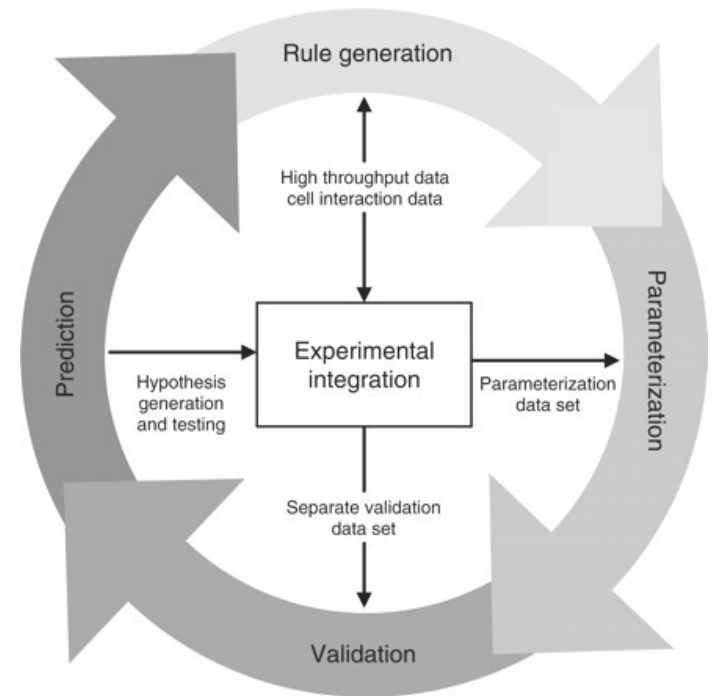
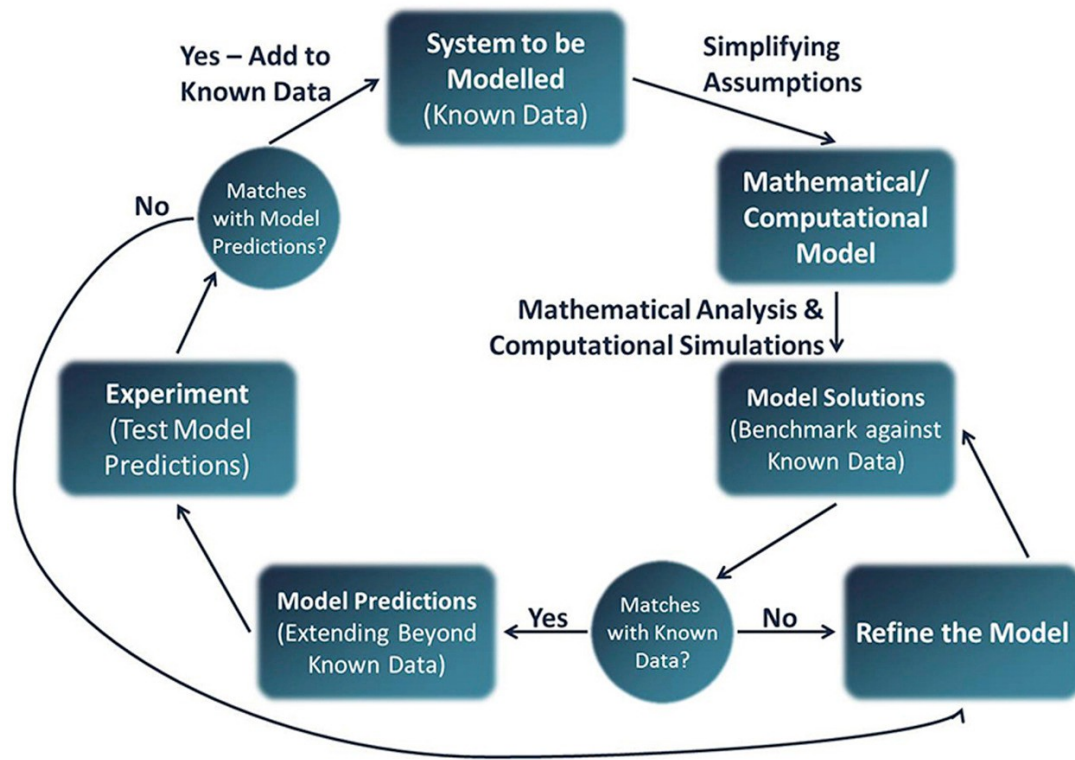
- Keywords used:
“agent-based modeling” or “agents-based simulation”



Important building blocks



The process of developing an ABM

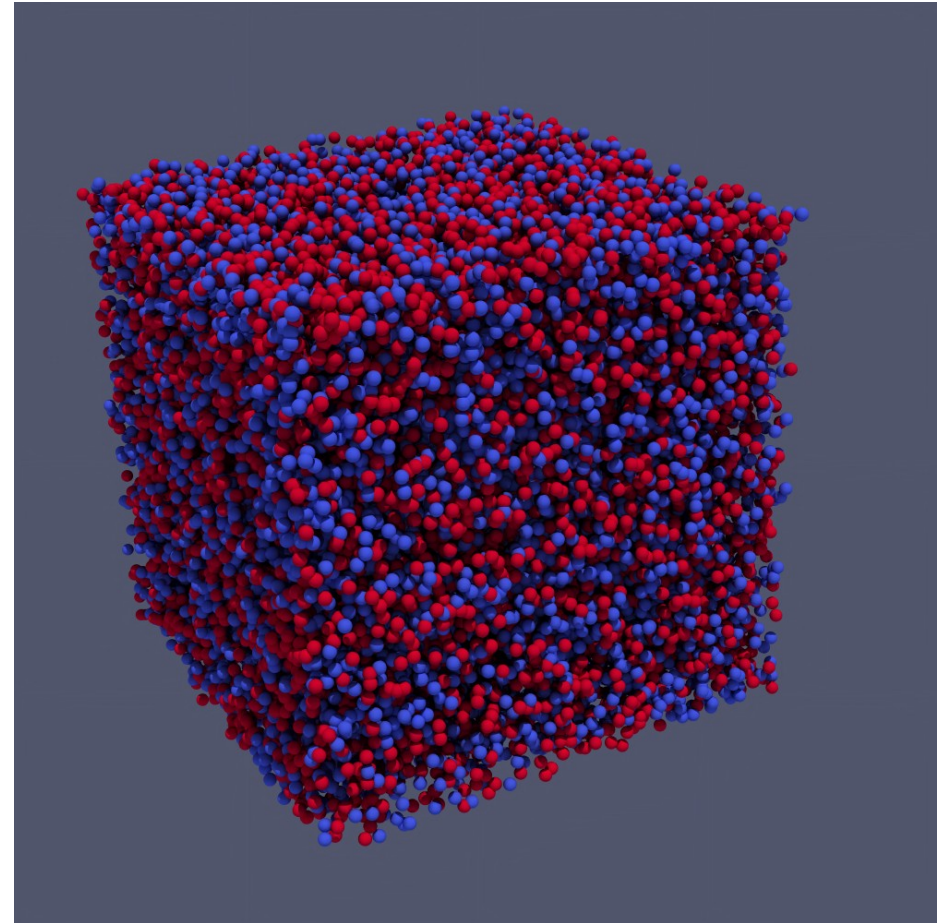




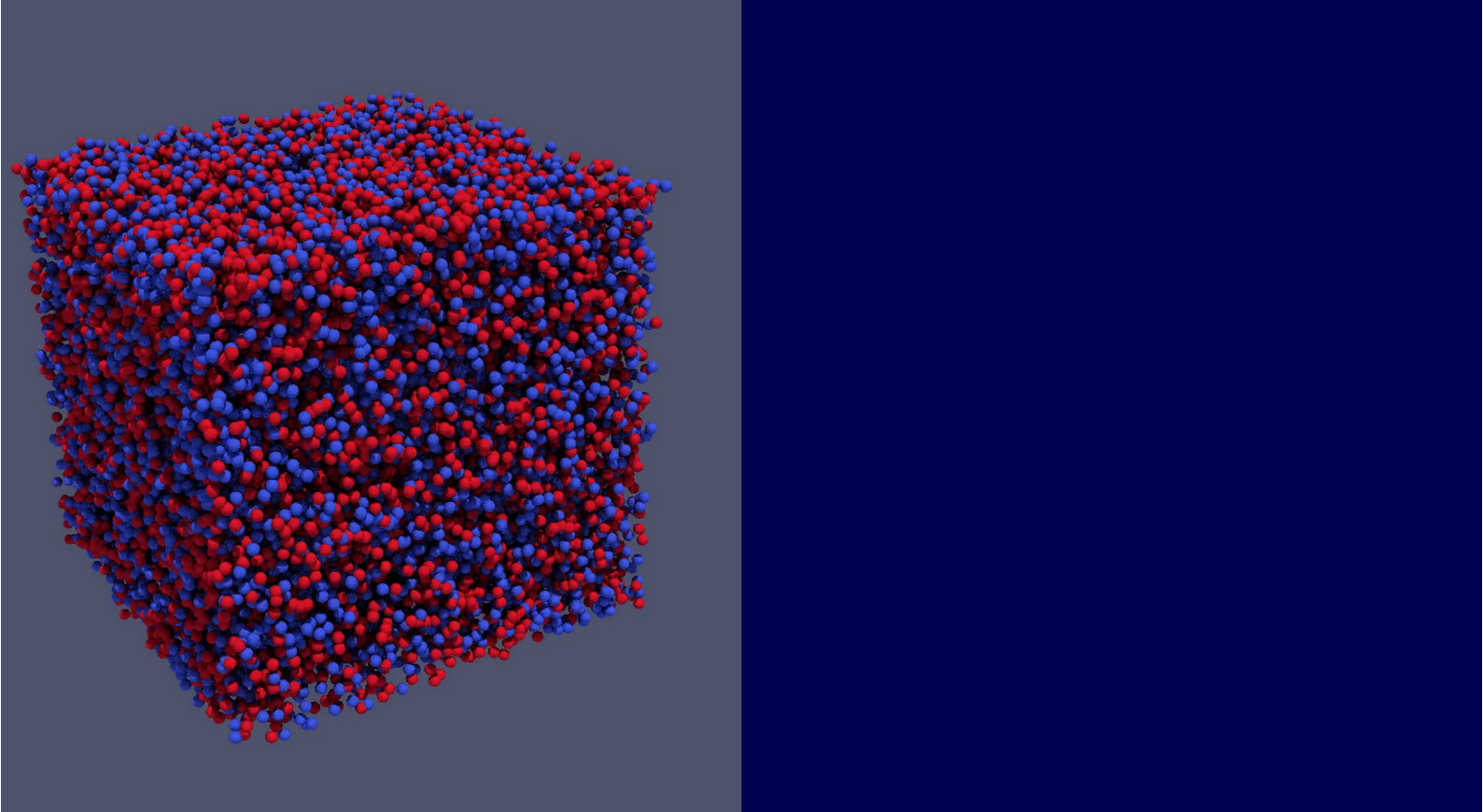
Demo

Cell clustering model

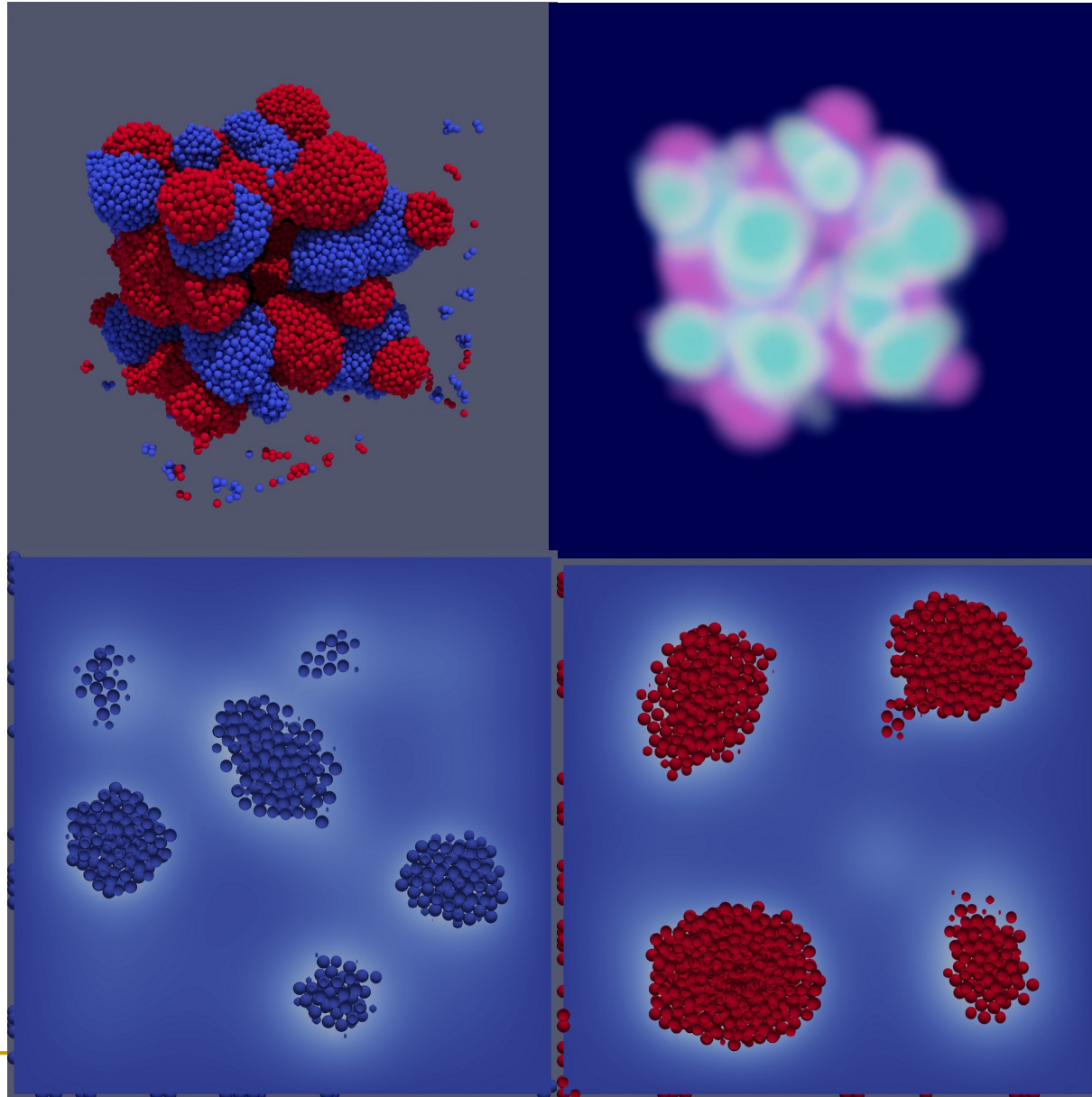
- Agent: Cell
 - Spherical shape
 - cell type
- Behaviors
 - Secrete a substance into the extracellular matrix
 - Follow the concentration gradient (chemotaxis)
- Initial condition
 - Randomly distributed in 3D space



Cell clustering video



Cell clustering result



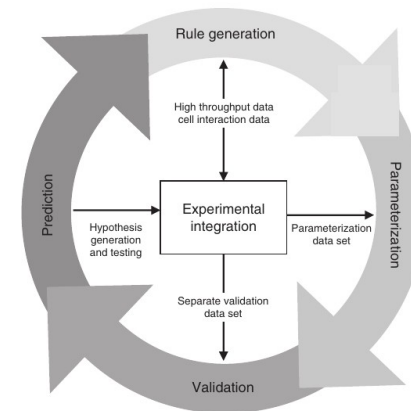
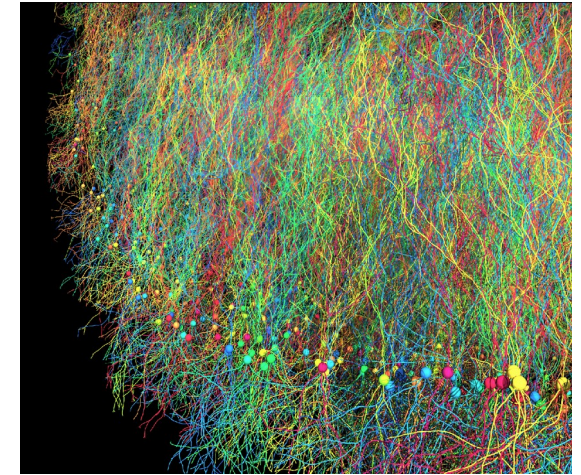
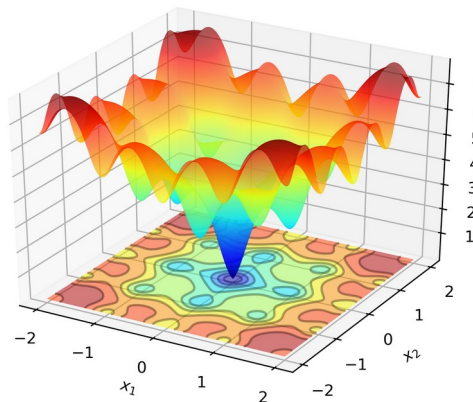
Performance considerations

The problem

Existing simulation platforms do not always take full advantage of modern hardware.

Impact of low performance

- Limitation of the size and complexity of models
- Longer development time
- Limited capability to explore parameter space
→ less optimal solution
- Increased cost



Our solution

BioDynaMo a **high-performance and modular, agent-based** simulation platform written in C++.

Features and abstraction layers

Simulation

- **Agent geometry:** sphere, cylinder
- **Agents:** Cell, NeuronSoma, NeuriteElement
- **Behaviors:** Secretion, Chemotaxis, Proliferation, GeneRegulation
- Extracellular diffusion
- Agent interaction force

Simulation

BioDynaMo's model building blocks

- Generation of agent populations
- Agent reproduction & mortality
- Environment search
- Multi-scale simulations
- Dynamic scheduling
- Statistical analysis
- Parameter management
- Parameter optimization
- Hierarchical model support
- Hybrid-modeling
- Space boundary conditions

BioDynaMo's high-level features

- Parallelism & thread-safety
- Performance optimizations
- GPU support
- Visualization
- Web-based interface
- Backup & restore of simulations
- Quality assurance infrastructure

BioDynaMo's low-level features

OpenMP

ROOT

ParaView

Others

Libraries

Linux / MacOS

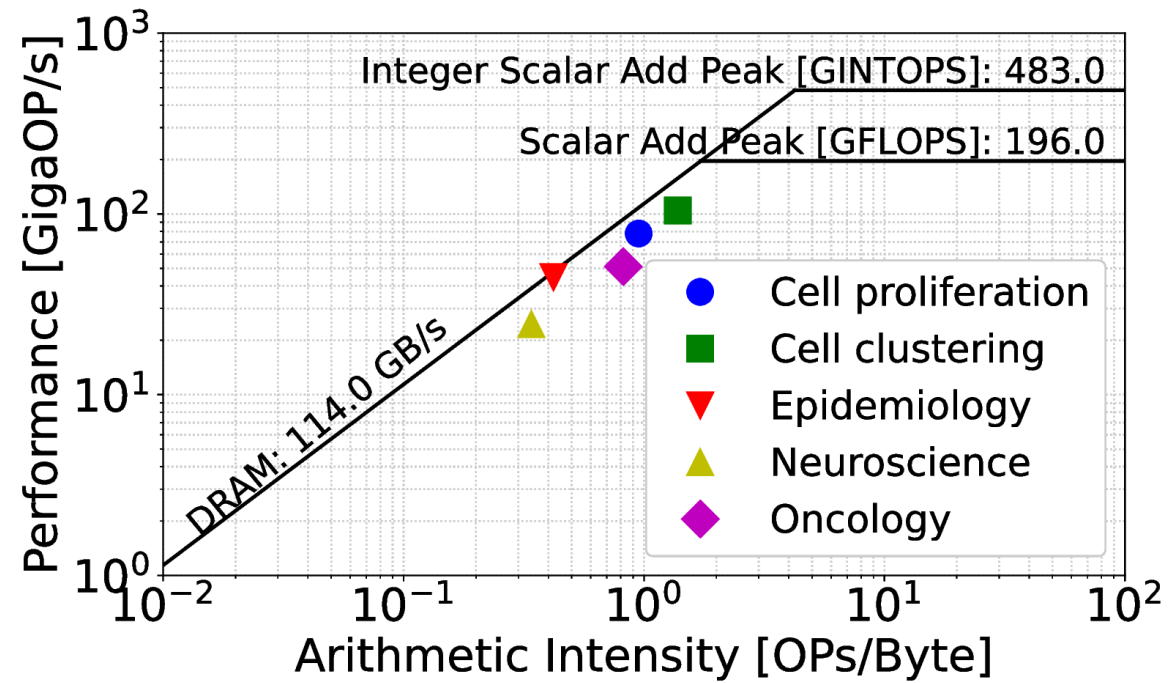
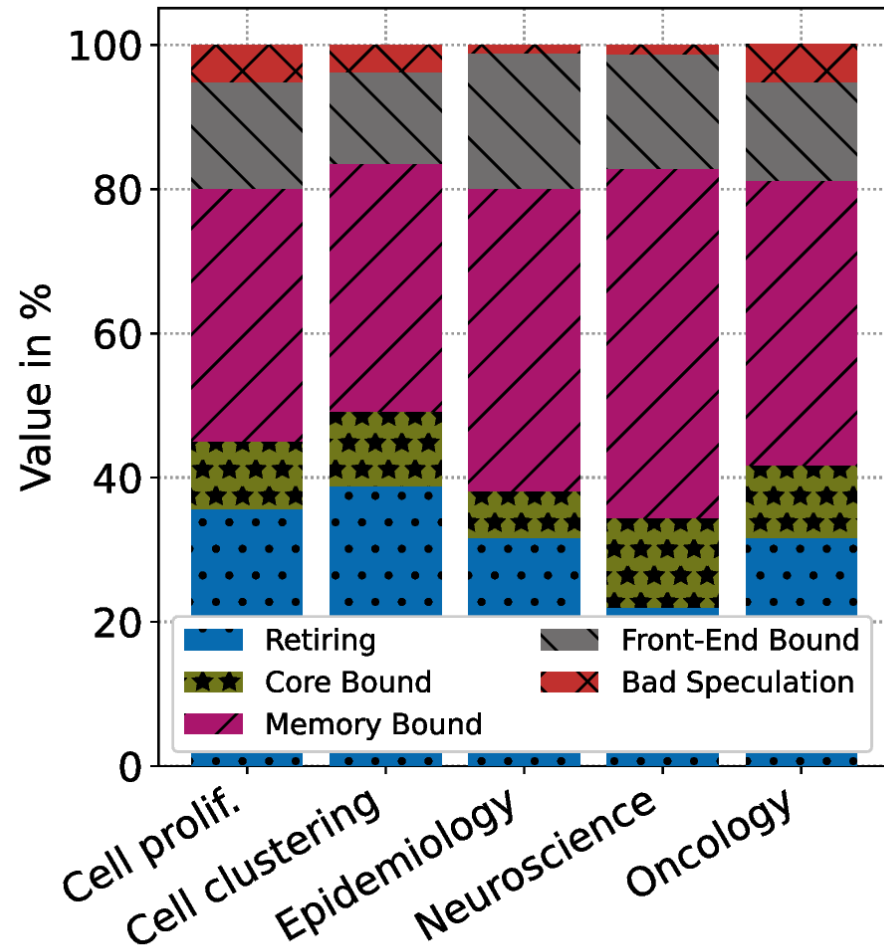
Operating System

(Multi-core) CPUs

GPU

Hardware

Challenge: Agent-based workload is memory-bound



Required efficient agent-based primitives

- Iterate over agents and execute operations
- Add and remove agents to/from the simulation
- Efficient environment implementation

Algorithm 1: Agent-based simulation algorithm

```
1 ModelInitialization()
2 for  $i \in iterations$  do
3   for  $op \in pre\_standalone\_operations$  do
4     |  $op()$ ;
5   end
6   wait()
7   parallel for  $a \in agents$  do
8     | for  $op \in agent\_operations$  do
9       | |  $op(a)$ ;
10    | end
11  end
12  for  $op \in standalone\_operations$  do
13    |  $op()$ ;
14  end
15  wait()
16  for  $op \in post\_standalone\_operations$  do
17    |  $op()$ ;
18  end
19 end
```

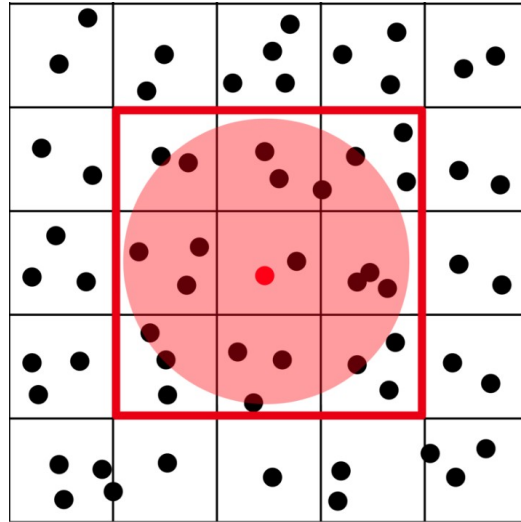
Performance Challenges and Improvements

- Maximize parallelization
 - Optimized uniform grid to search for neighbors
 - Parallelize the addition and removal of agents
- Efficient thread synchronization during agent updates
- Minimize memory access latency
 - NUMA-aware iteration
 - Agent Sorting and Balancing
 - Pool-based memory allocator
- Offload computation to the GPU

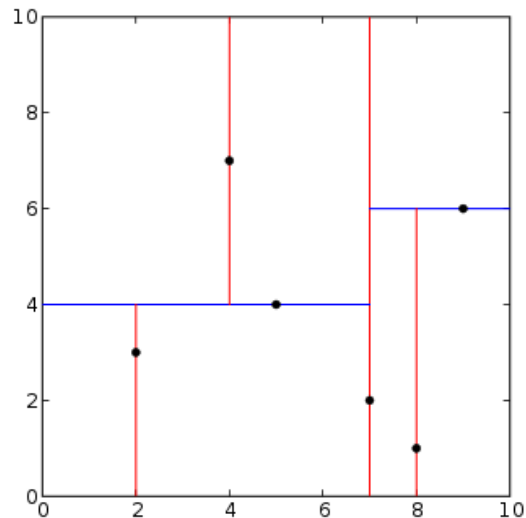
Maximize parallelization

Optimized uniform grid to search for neighbors

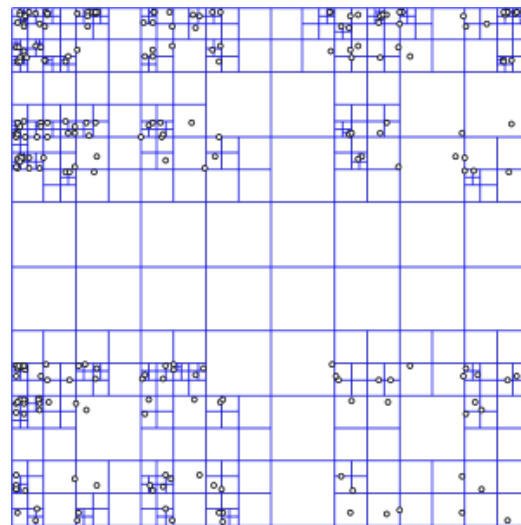
Uniform Grid



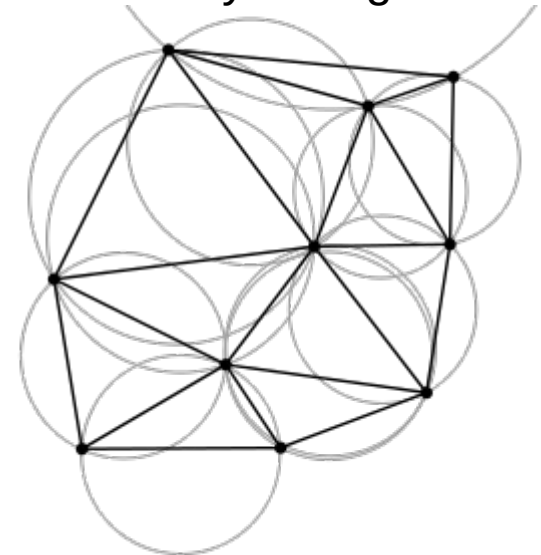
Kd-Tree



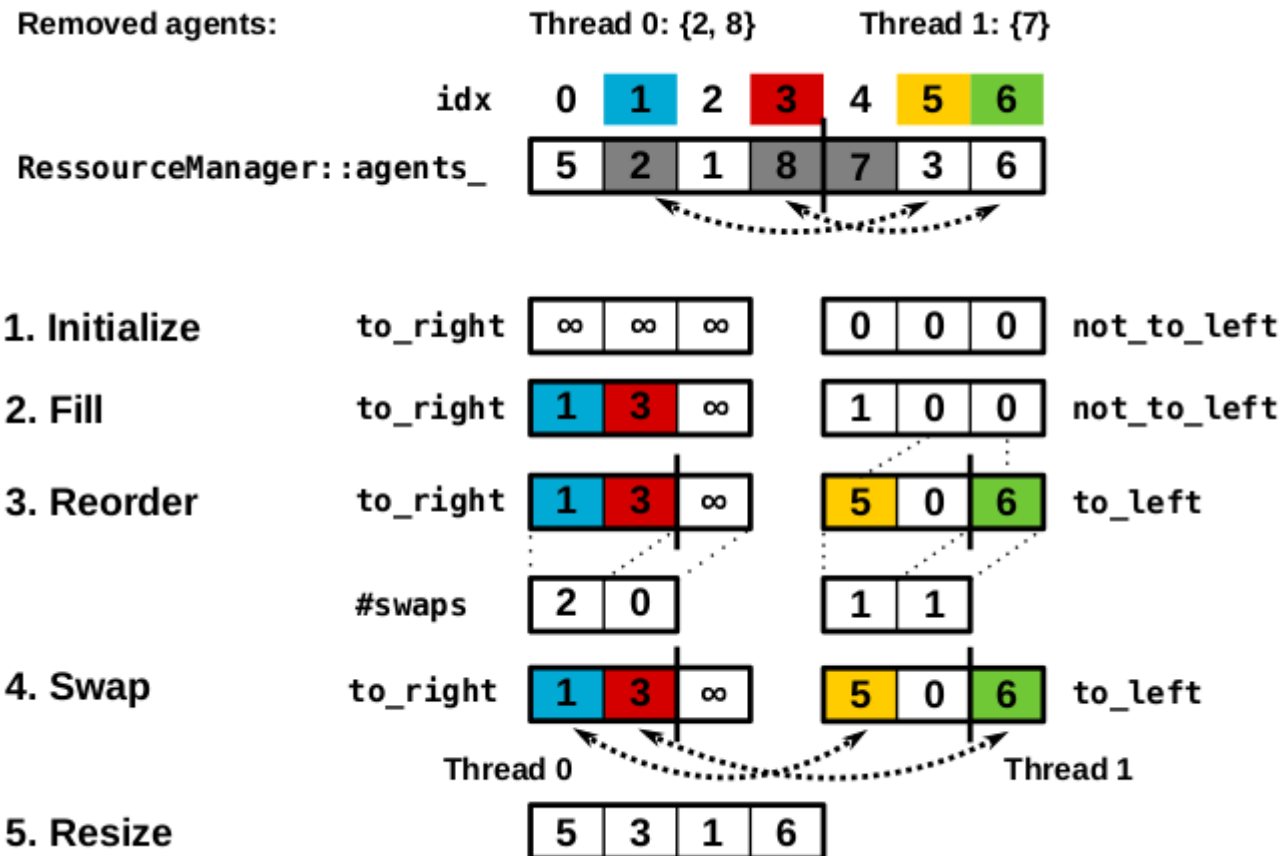
Quadtree



Delaunay Triangulation



Parallel agent removal mechanism

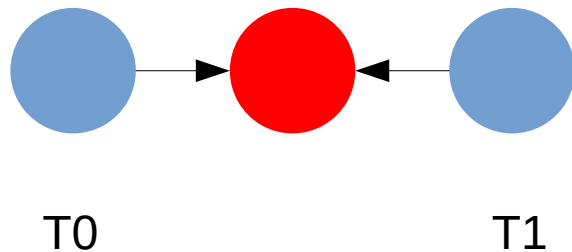


Optimize Thread-Synchronization

Thread-synchronization (TS) during agent-updates

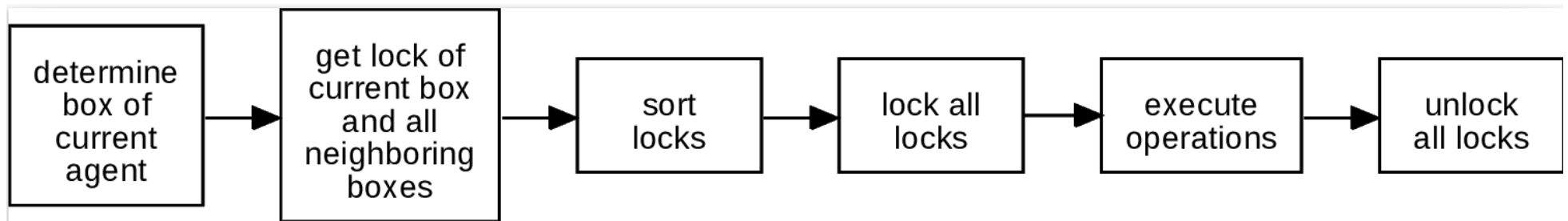
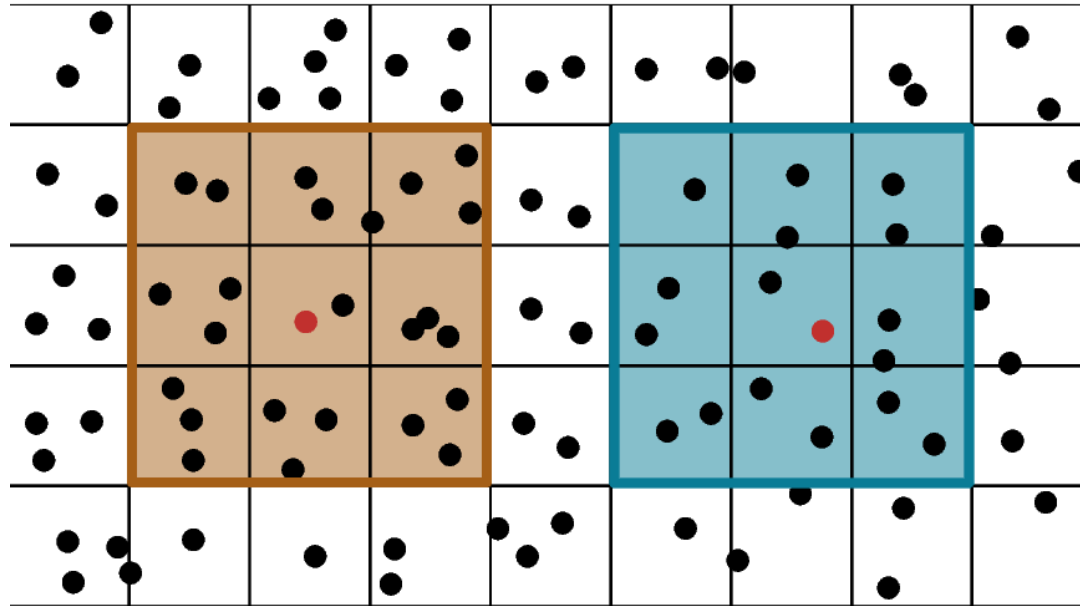
Algorithm 1: Agent-based simulation algorithm

```
1 ModelInitialization()
2 for  $i \in \text{iterations}$  do
3   for  $op \in \text{pre\_standalone\_operations}$  do
4     |  $op()$ ;
5   end
6   wait()
7   parallel for  $a \in \text{agents}$  do
8     for  $op \in \text{agent\_operations}$  do
9       |  $op(a)$ ;
10    end
11  end
12  for  $op \in \text{standalone\_operations}$  do
13    |  $op()$ ;
14  end
15  wait()
16  for  $op \in \text{post\_standalone\_operations}$  do
17    |  $op()$ ;
18  end
19 end
```

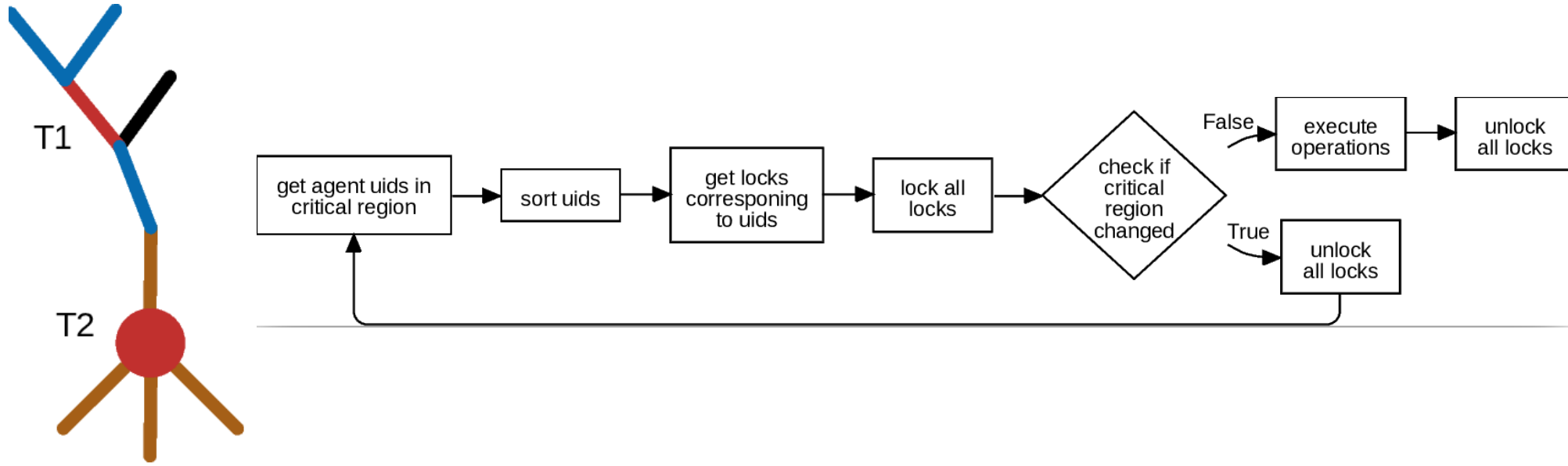


- Only necessary if agents modify their local environment.
 - Two agents (updated by two different threads) could attempt to modify the same neighbor.
- BioDynaMo provides two TS mechanisms
 - Automatic TS
 - User-defined TS

Automatic thread-synchronization



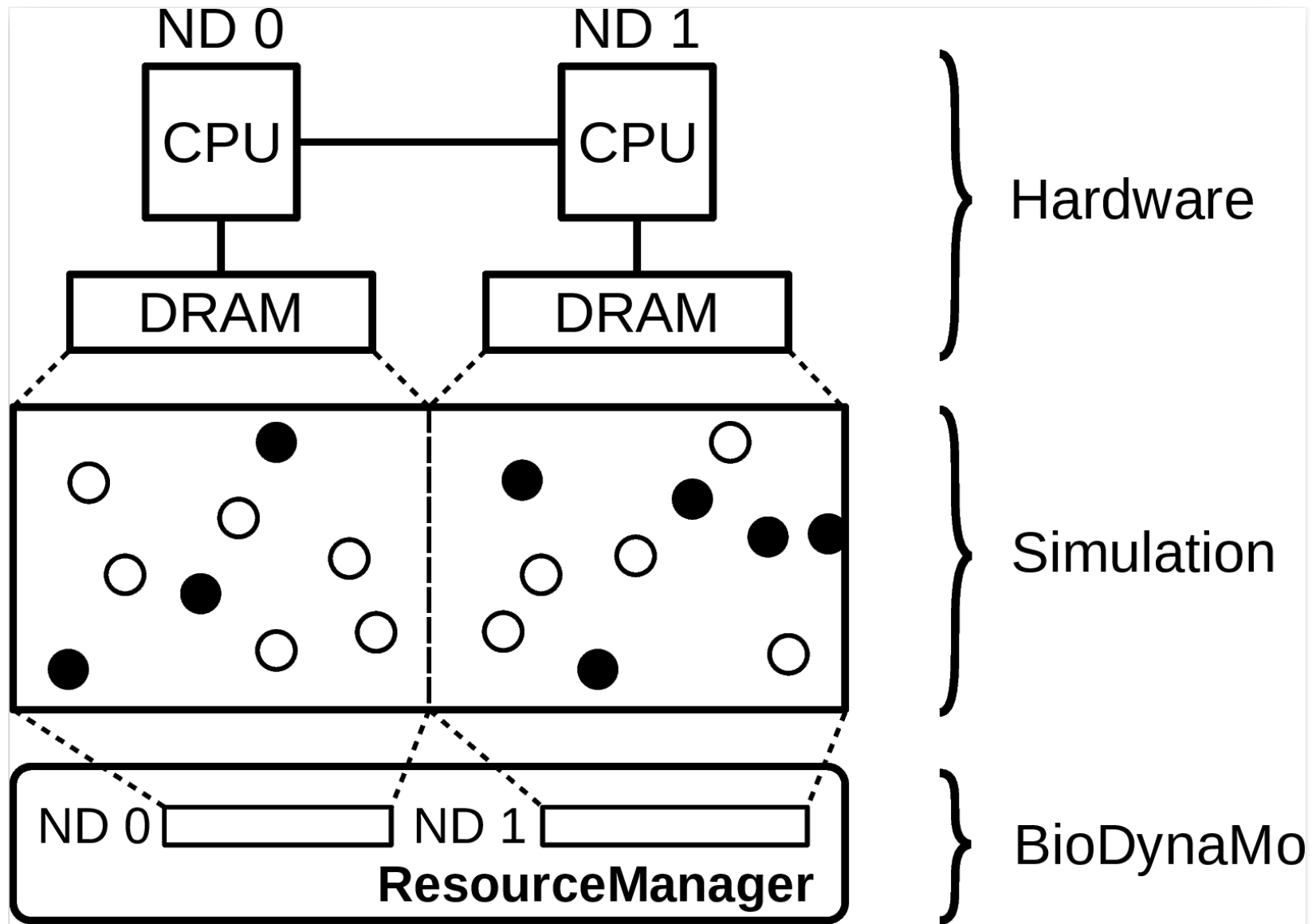
User-defined thread-synchronization



```
void NeuronSoma::CriticalRegion(std::vector<AgentPointer<>>* aptrs) {
    aptrs->reserve(daughters_.size() + 1);
    aptrs->push_back(Agent::GetAgentPtr<>());
    for (auto& daughter : daughters_) {
        aptrs->push_back(daughter);
    }
}
```

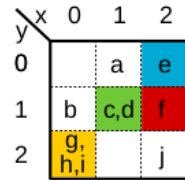
Minimize Memory Access Latency

NUMA-aware iteration

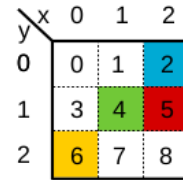


Agent sorting and balancing mechanism

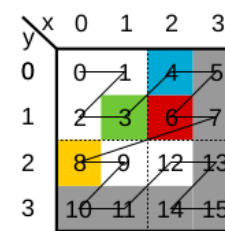
A Agents in 3x3 grid



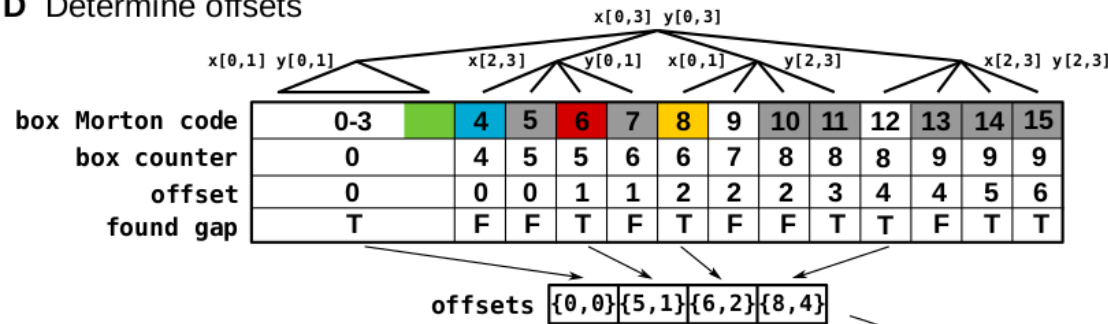
B Grid box indices



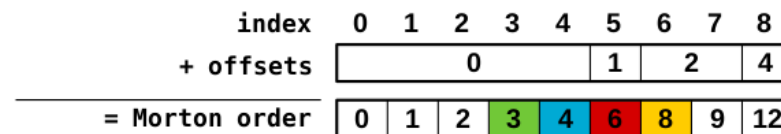
C Morton order of 4x4 grid



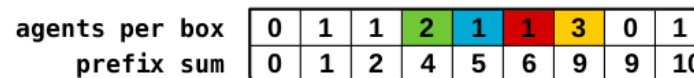
D Determine offsets



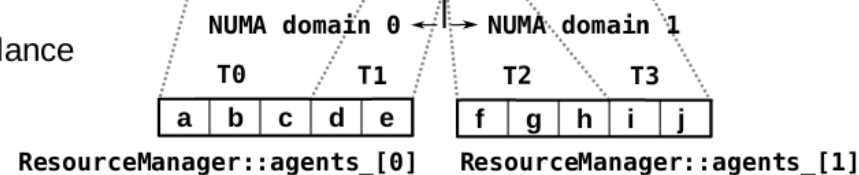
E Determine Morton order



F Partition

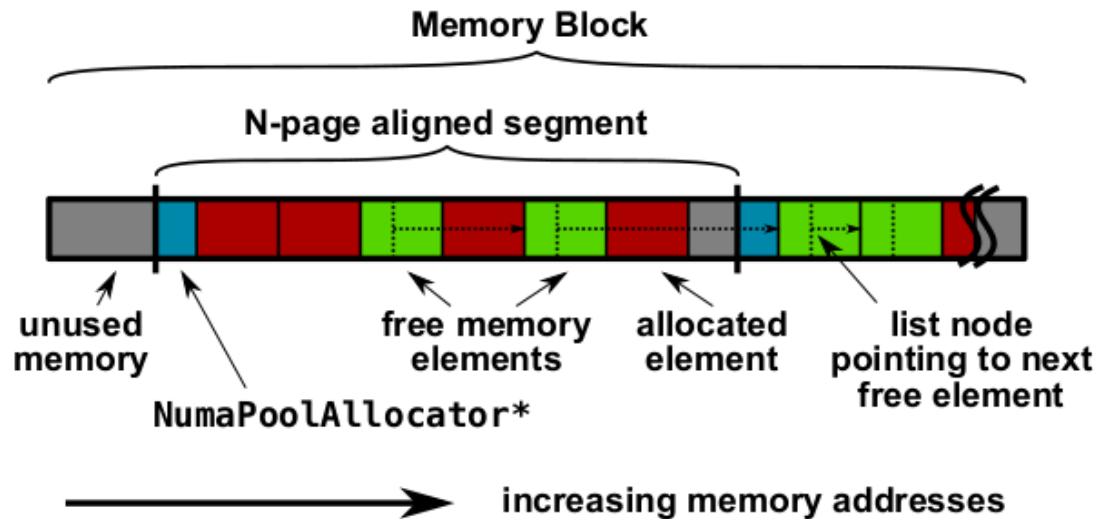


G Sort and balance

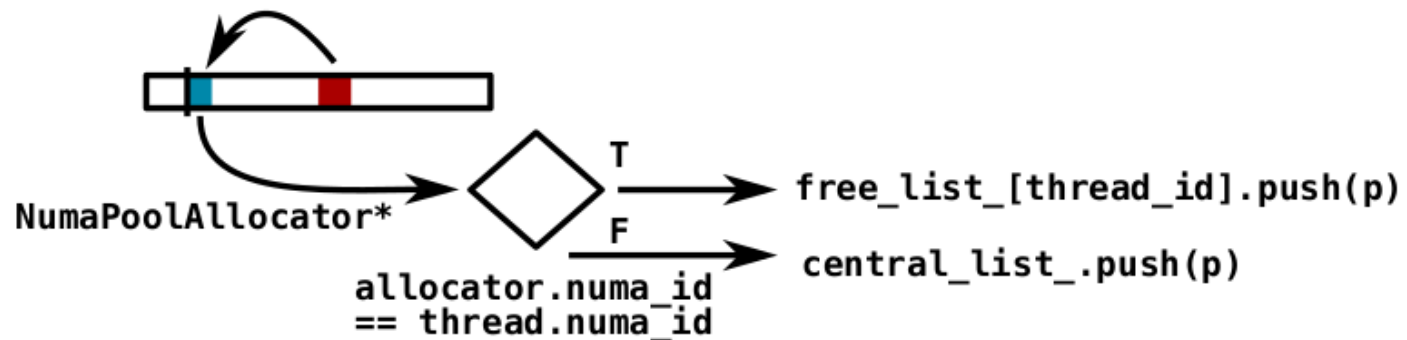


BioDynaMo Memory Allocator

A Layout



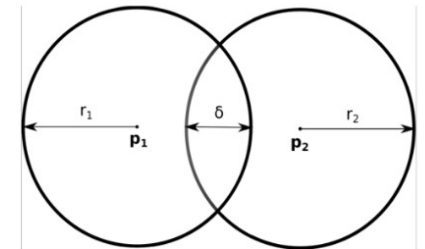
B Deallocation



Offload computation to the GPU

BioDynaMo's GPU capabilities

- Operations can have implementations for different compute targets (CPU, GPU, and FPGA).
- If an operation has multiple implementations, the scheduler decides which one to use.
- Currently, BioDynaMo provides a GPU operation to calculate mechanical forces between spheres.

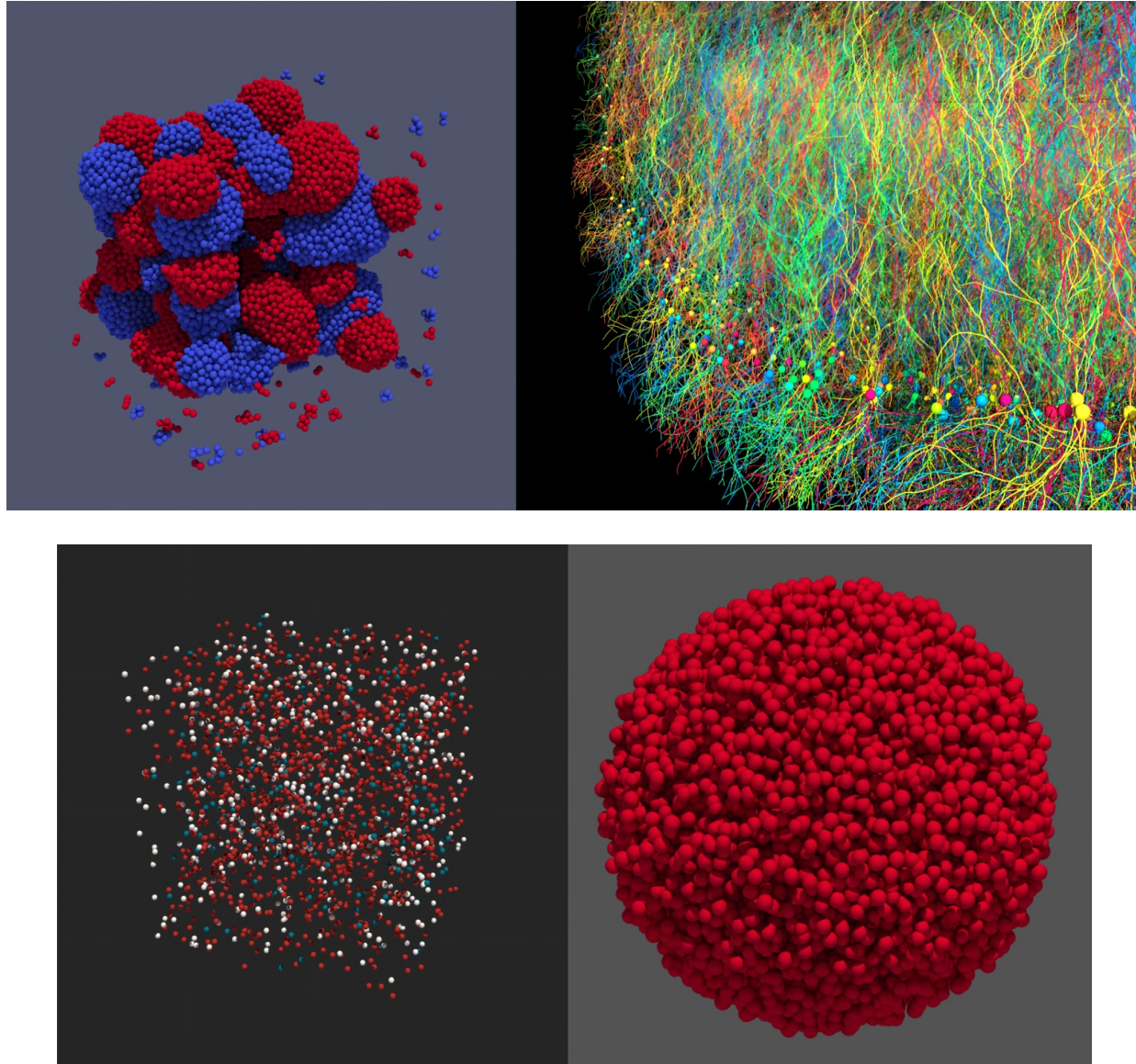


$$\begin{aligned}\delta &= r_1 + r_2 - \|p_1 - p_2\| \\ r &= \frac{r_1 \cdot r_2}{r_1 + r_2} \\ F &= (\kappa \cdot \delta - \gamma \cdot \sqrt{r \cdot \delta}) \cdot \frac{p_1 - p_2}{\|p_1 - p_2\|}\end{aligned}$$

Collision force computation

Performance Evaluation

Benchmark simulations



Benchmark simulations characteristics

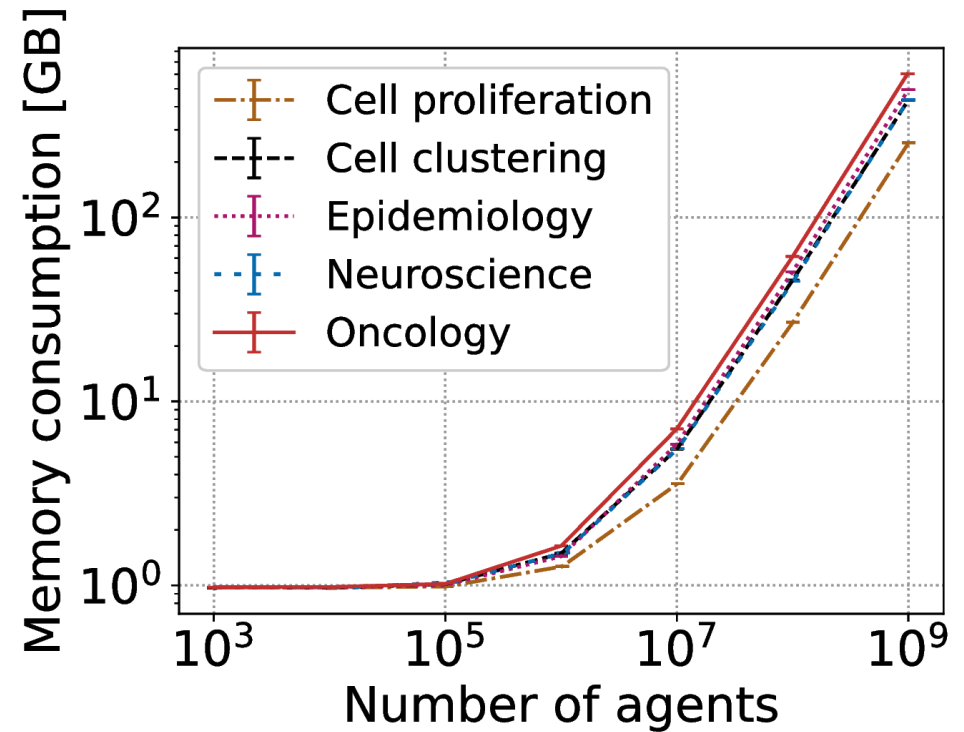
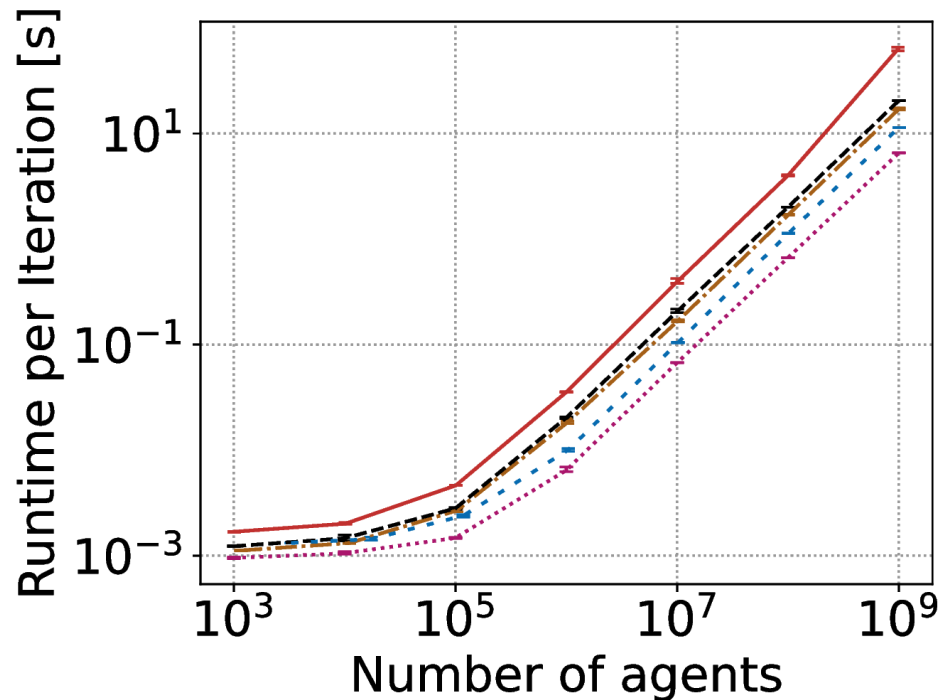
Characteristic	Cell proliferation	Cell clustering	Epidemiology use case	Neuroscience use case	Oncology use case
Create new agents during simulation	X			X	X
Delete agents during simulation					X
Agents modify neighbors				X	
Load imbalance			X	X	
Agents move randomly			X		X
Simulation uses diffusion		X		X	
Simulation has static regions				X	
Number of iterations	500	1000	1000	500	288
Number of agents (in millions)	12.6	2	10	9	10
Number of diffusion volumes	0	54m	0	65k	0

Benchmark hardware

TABLE II: Benchmark hardware

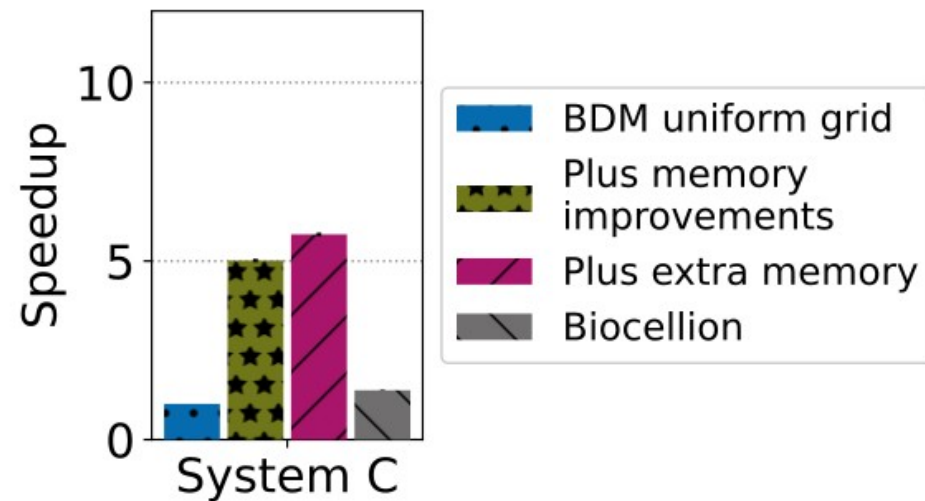
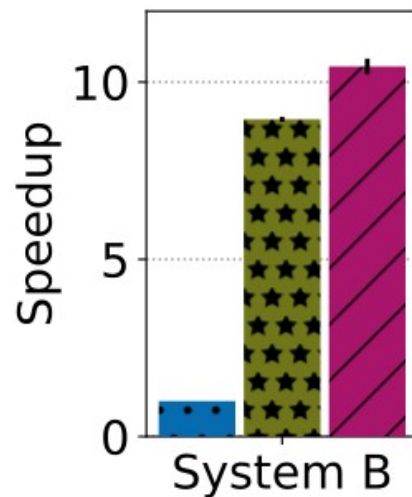
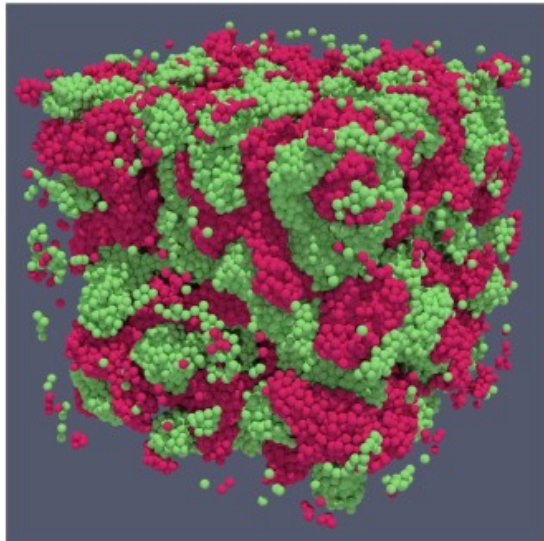
System	Main memory	CPU	OS
A	504 GB	Server with four Intel(R) Xeon(R) E7-8890 v3 CPUs @ 2.50GHz with a total of 72 physical cores, two threads per core and four NUMA nodes.	CentOS 7.9.2009
B	1008 GB		
C	62 GB	Server with two Intel(R) Xeon(R) E5-2683 v3 CPUs @ 2.00GHz with a total of 28 physical cores, two threads per core and two NUMA nodes.	CentOS Stream 8

Runtime and Memory Complexity

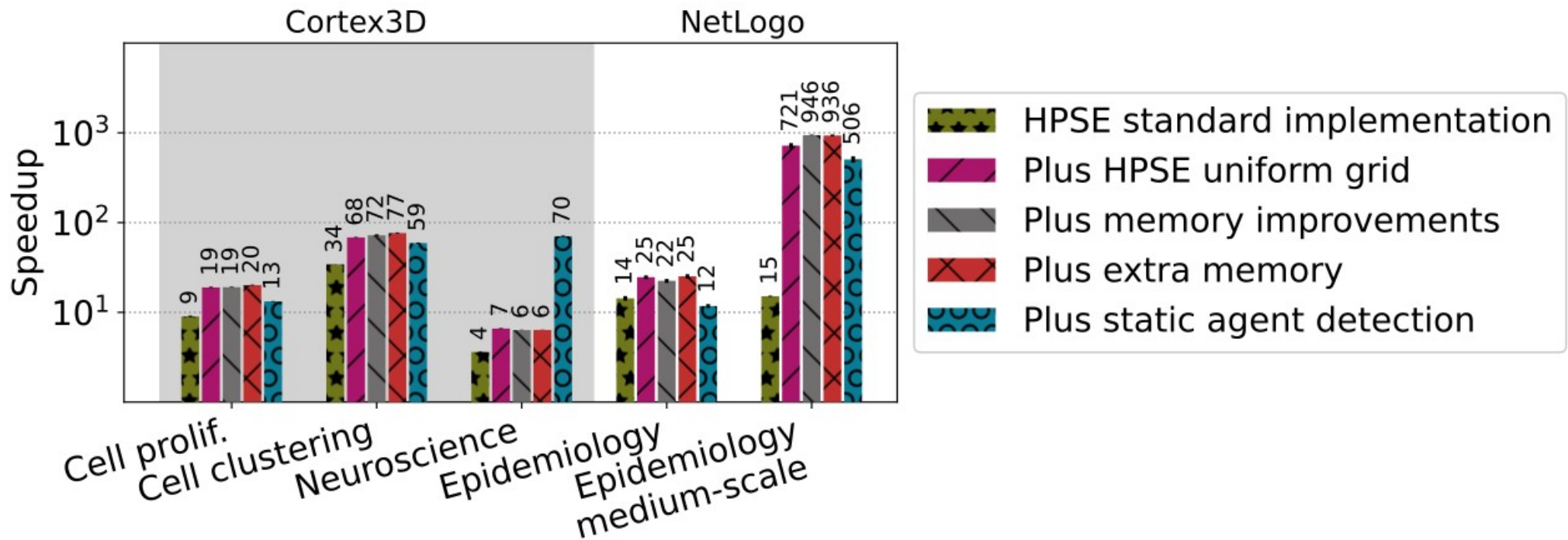


Comparison with Biocellion

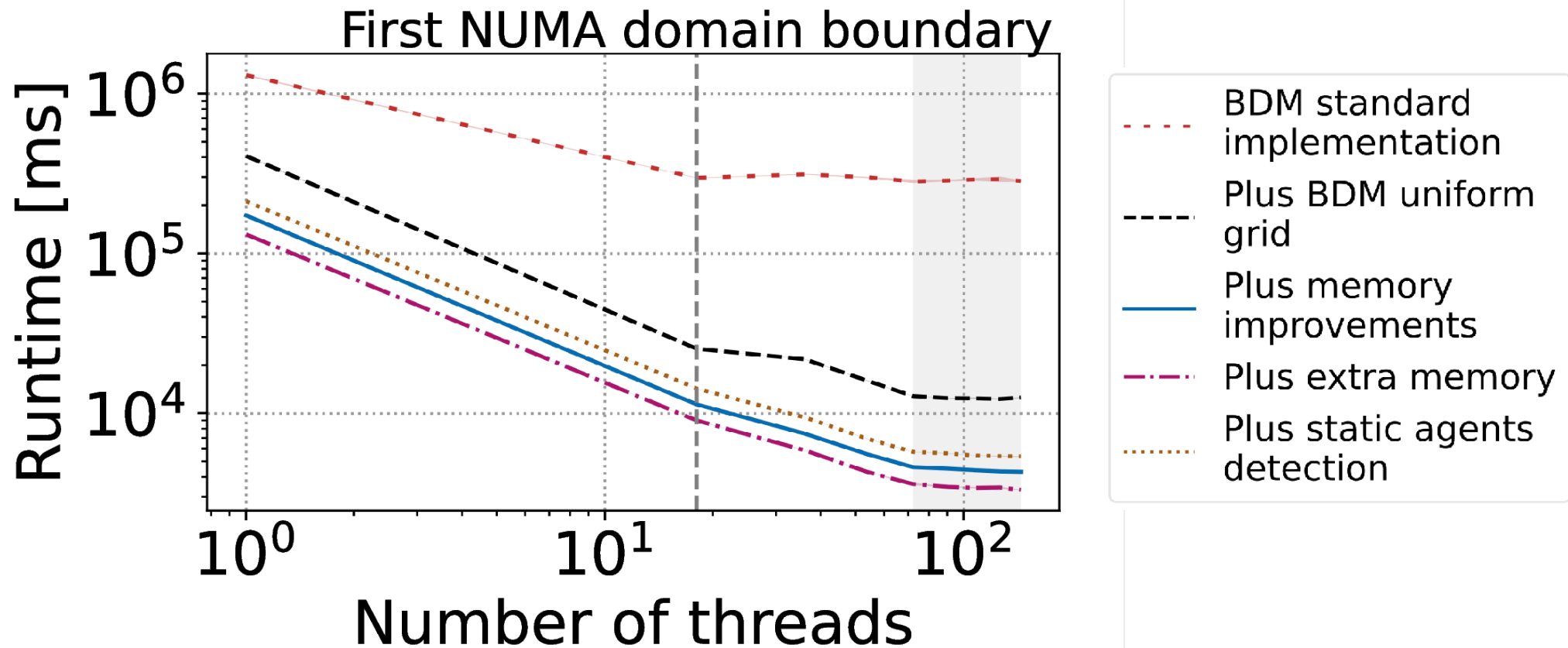
- Single-node 16 CPU cores; 13.4 million cells
→ BioDynaMo is **4.15x faster**
- Biocellion: 21 nodes, 672 CPU cores, 281 million cells
BioDynaMo: one node, 72 CPU cores
→ same runtime, but **9.3x fewer CPU cores** used



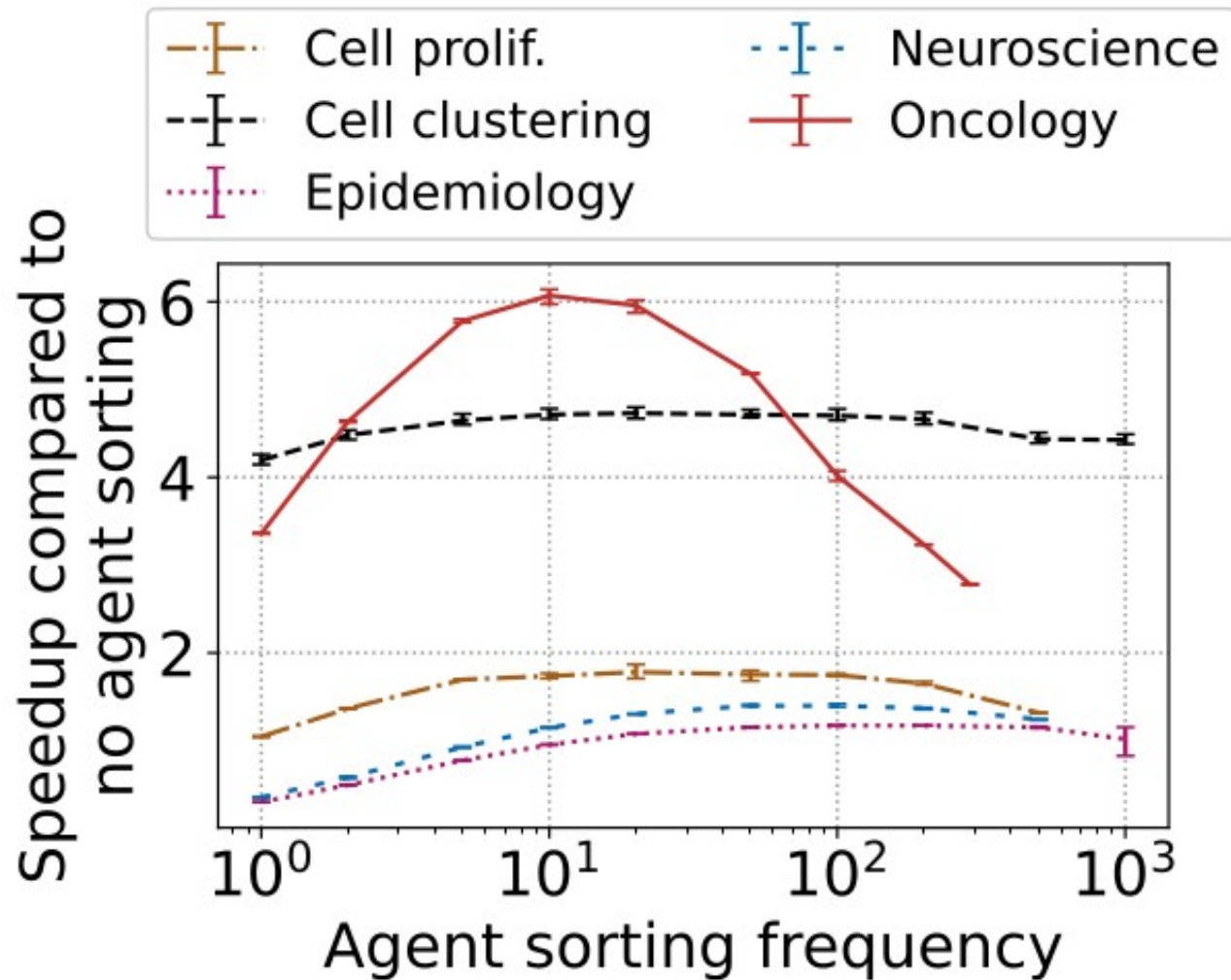
Comparison with Cortex3D and NetLogo



Scalability



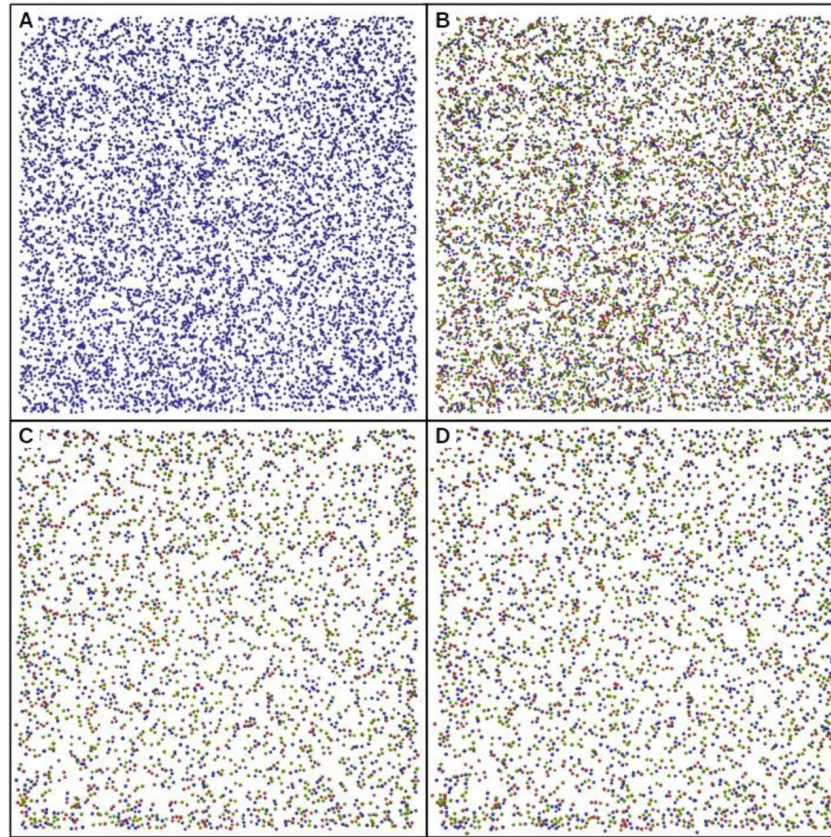
Agent Sorting and Balancing



Ongoing Use Cases

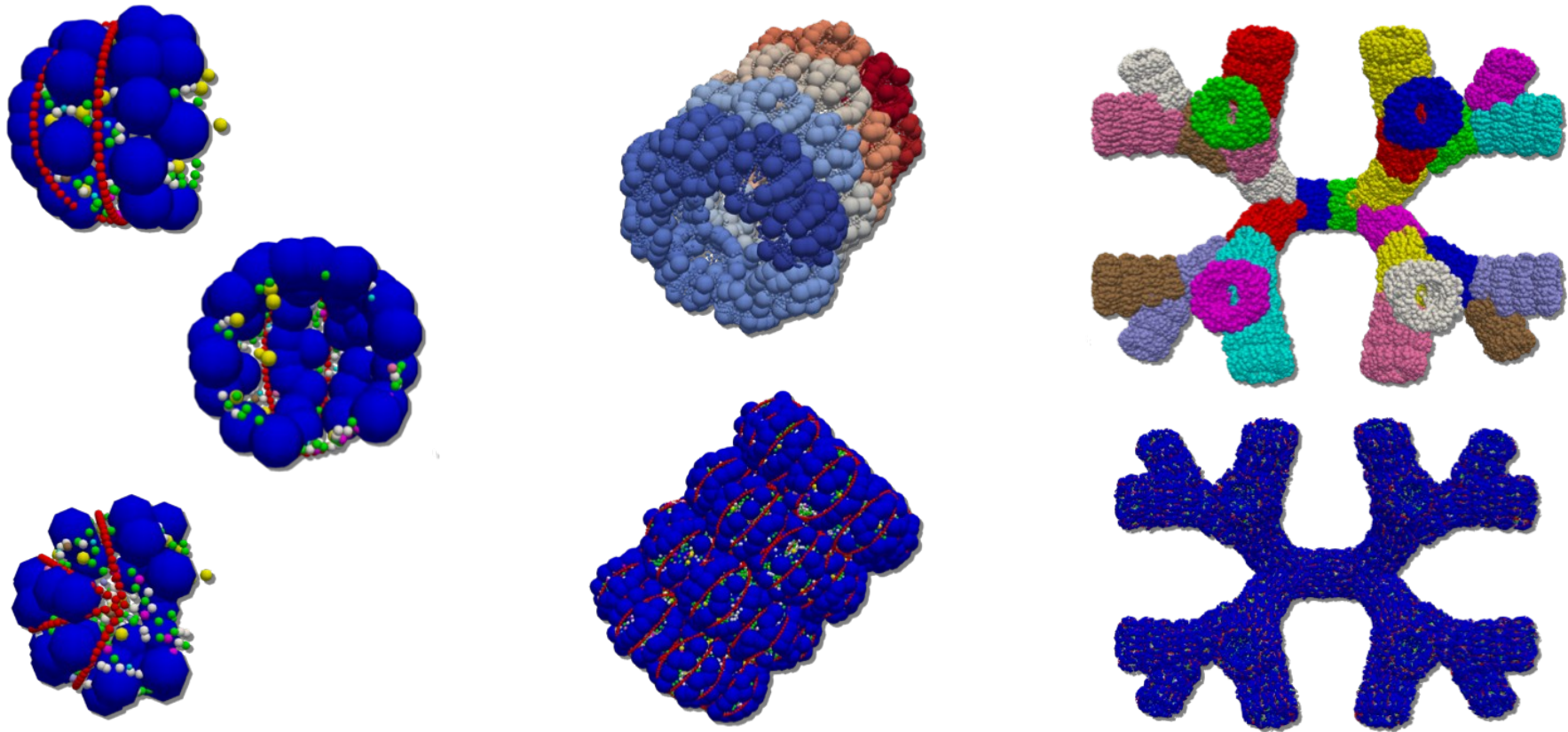
Retinal self-organization

- Understand the mechanisms of cells self-organization during early development which is pivotal for their function.



Radiation-induced lung injury simulation


- Simulate onset of radiation pneumonitis and/or lung fibrosis in normal tissue after exposition to thoracic irradiation.



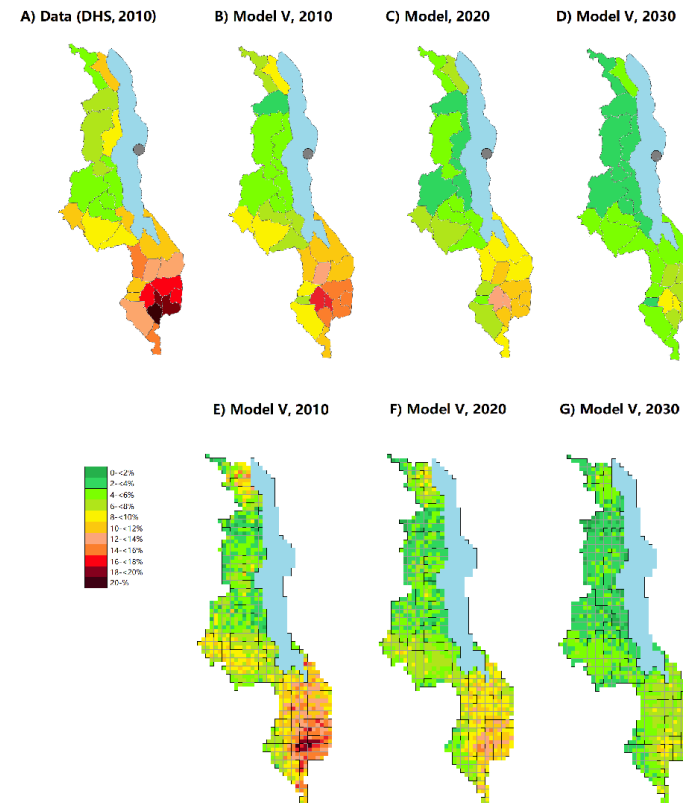
Spatial Spread of HIV in Malawi

- Collaboration with UniGE
- Original simulation written in R (Runtime: ~5.5h)
- Goal: speed up execution time
- Preliminary runtime with BioDynaMo: less than **2 minutes**
- Further work needed to make models equivalent

The spatial spread of HIV in Malawi: An individual-based mathematical model

 Janne Estill, Wingston Ng'ambi, Liudmila Rozanova, Olivia Keiser

doi: <https://doi.org/10.1101/2020.12.23.20248757>



Summary

- Agent-based simulation can be used to model many systems
- The presented optimizations improve the performance over state-of-the-art
 - Up to **three orders of magnitude speedup**
- These improvements allow BioDynaMo simulating **billions of agents** on a single server
- BioDynaMo is currently being used in:
 - neuroscience
 - oncology
 - epidemiology
 - cryobiology
 - socioeconomics
 - finance
 - ...

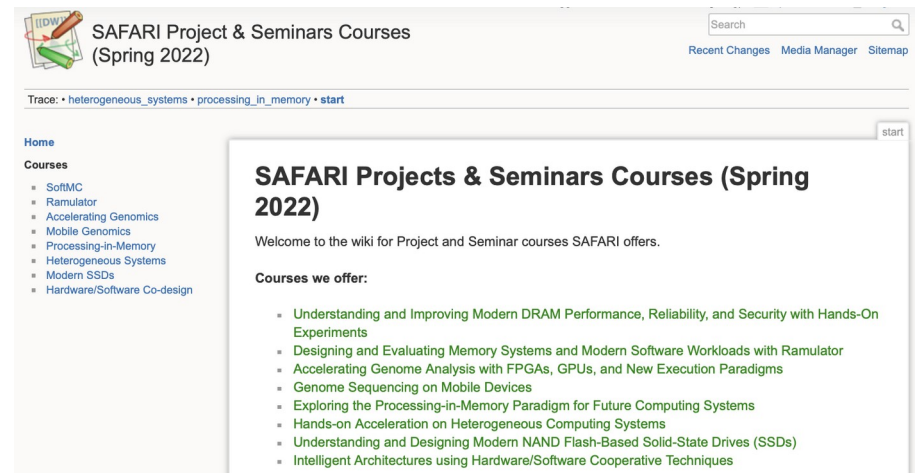
Thank you for your attention!

Lukas.Breitwieser@cern.ch

More P&S Courses: SSDs, Memory, Bioinformatics...

- Understanding and Improving Modern DRAM Performance, Reliability, and Security with Hands-On Experiments
- Designing and Evaluating Memory Systems and Modern Software Workloads with Ramulator
- Accelerating Genome Analysis with FPGAs, GPUs, and New Execution Paradigms
- Genome Sequencing on Mobile Devices
- Understanding and Designing Modern NAND Flash-Based Solid-State Drives (SSDs)
- Intelligent Architectures using Hardware/Software Cooperative Techniques

https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=start



SAFARI Project & Seminars Courses (Spring 2022)

Trace: • heterogeneous_systems • processing_in_memory • start

Home

Courses

- SoftMC
- Ramulator
- Accelerating Genomics
- Mobile Genomics
- Processing-in-Memory
- Heterogeneous Systems
- Modern SSDs
- Hardware/Software Co-design

SAFARI Projects & Seminars Courses (Spring 2022)

Welcome to the wiki for Project and Seminar courses SAFARI offers.

Courses we offer:

- Understanding and Improving Modern DRAM Performance, Reliability, and Security with Hands-On Experiments
- Designing and Evaluating Memory Systems and Modern Software Workloads with Ramulator
- Accelerating Genome Analysis with FPGAs, GPUs, and New Execution Paradigms
- Genome Sequencing on Mobile Devices
- Exploring the Processing-in-Memory Paradigm for Future Computing Systems
- Hands-on Acceleration on Heterogeneous Computing Systems
- Understanding and Designing Modern NAND Flash-Based Solid-State Drives (SSDs)
- Intelligent Architectures using Hardware/Software Cooperative Techniques

More Resources: Onur Mutlu Lectures

- All P&S courses
- Digital Design and CompArch course
- Advanced CompArch course
- Seminar in CompArch

The screenshot displays the YouTube channel page for "Onur Mutlu Lectures", which has 25.6K subscribers. The page is organized into several sections:

- Channel Header:** Includes the channel name, subscriber count, and navigation tabs (HOME, VIDEOS, PLAYLISTS, COMMUNITY, CHANNELS, ABOUT).
- Created playlists:** A row of six featured playlists with video thumbnails and titles:
 - SAFARI Live Seminars 2022:** Updated 4 days ago.
 - Livestream - P&S Modern SSDs (Spring 2022):**
 - Livestream - P&S Intelligent Architectures via...:**
 - Livestream - P&S Hands-on Acceleration on Heterogeneous...:** Updated 6 days ago.
 - Livestream - P&S Exploring the Processing-in-Memory...:** Updated 6 days ago.
 - Livestream - P&S Accelerating Genome Analysis with FPGAs...:** Updated 7 days ago.
- First Course in Computer Architecture & Digital Design 2021-2013:** A row of six playlists:
 - Livestream - Digital Design and Computer Architecture - ETH...:**
 - Digital Design & Computer Architecture - ETH Zürich...:**
 - Design of Digital Circuits - ETH Zürich - Spring 2019:**
 - Design of Digital Circuits - ETH Zürich - Spring 2018:**
 - Digital Circuits and Computer Architecture - ETH Zurich - ...:**
 - Spring 2015 -- Computer Architecture Lectures --...:**
- Advanced Computer Architecture Courses 2021-2012:** A row of six playlists, including titles like "Processing in DRAM Engine" and "ML acceleration: 260 mm², 6 billion 600 GFLOPS GPU, 12 ARM 2.2 GHz".

P&S Heterogeneous Systems

Accelerating Agent-Based Simulations

Lukas Breitwieser

ETH Zürich

Spring 2022

11 July 2022