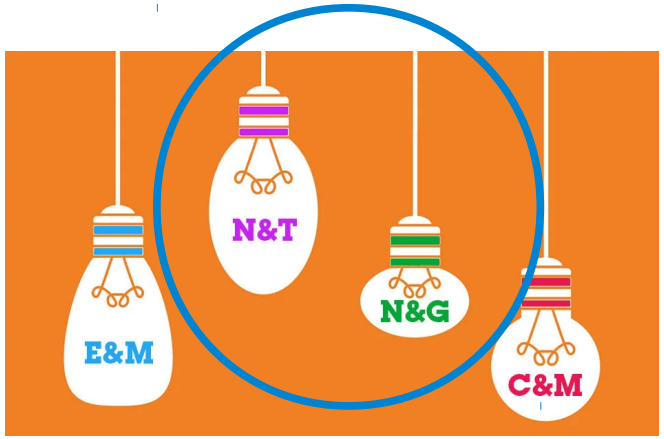
CERN openlab

Biomedische Simulaties op CERN

Dutch Teacher's Programme

Ahmad Hesam
26 September 2019

Introductie



- Tweetalig VWO gestudeerd in Apeldoorn
- Natuur & Techniek + Natuur & Gezondheid
- Technische Natuurkunde (Applied Physics)
- 3e jaar: hmm… wil ik dit blijven doen?
- Interesse in Quantum Computers heeft me overtuigd om een Minor in Computer Science / Engineering te doen

TU Delft

Universiteit: Master

- Minor was leuk: ik doe de master!
- High-Performance Computing is de (nabije) toekomst
- Delftse Bedrijvendagen: CERN kwam langs

  → CERN biedt zomerstages aan voor studenten!

CERN openlab summer programme

- 9 weken in de zomer
- Betaalde stage
- Lezingen over CERN technologiën
- Mogelijkheden om te blijven
- Gemiddeld 40 studenten uit 20 verschillende landen

CERN openlab

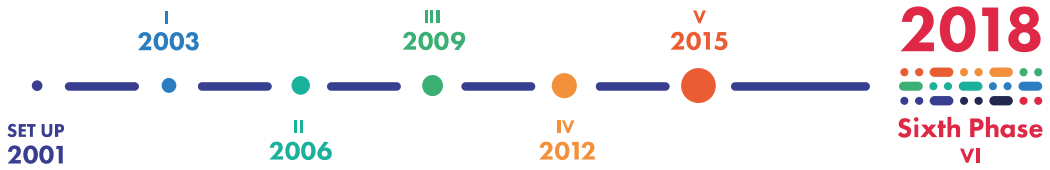
A science – industry partnership to drive R&D and innovation

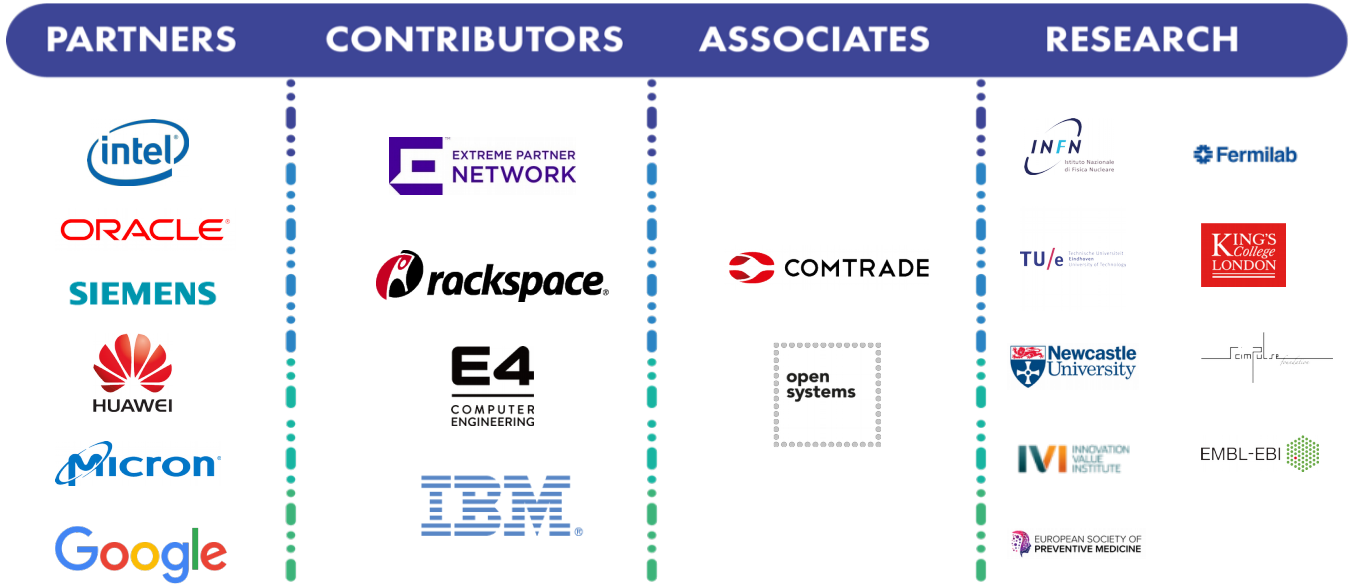
Evaluate **state-of-the-art technologies** in a challenging environment and improve them

Test in a **research environment** today what will be used in many business sectors tomorrow

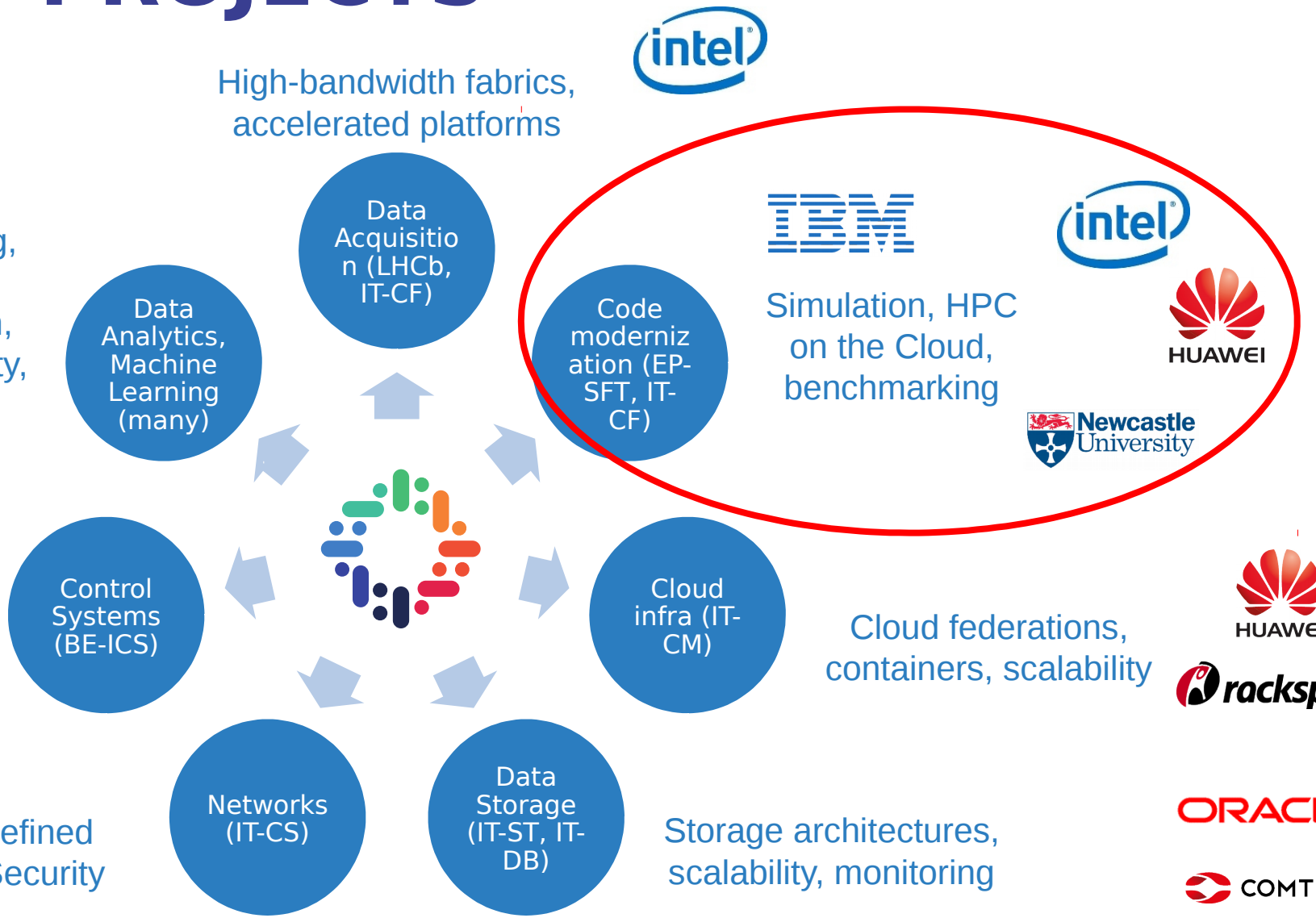
Training

Dissemination and **outreach**





JOINT R&D PROJECTS

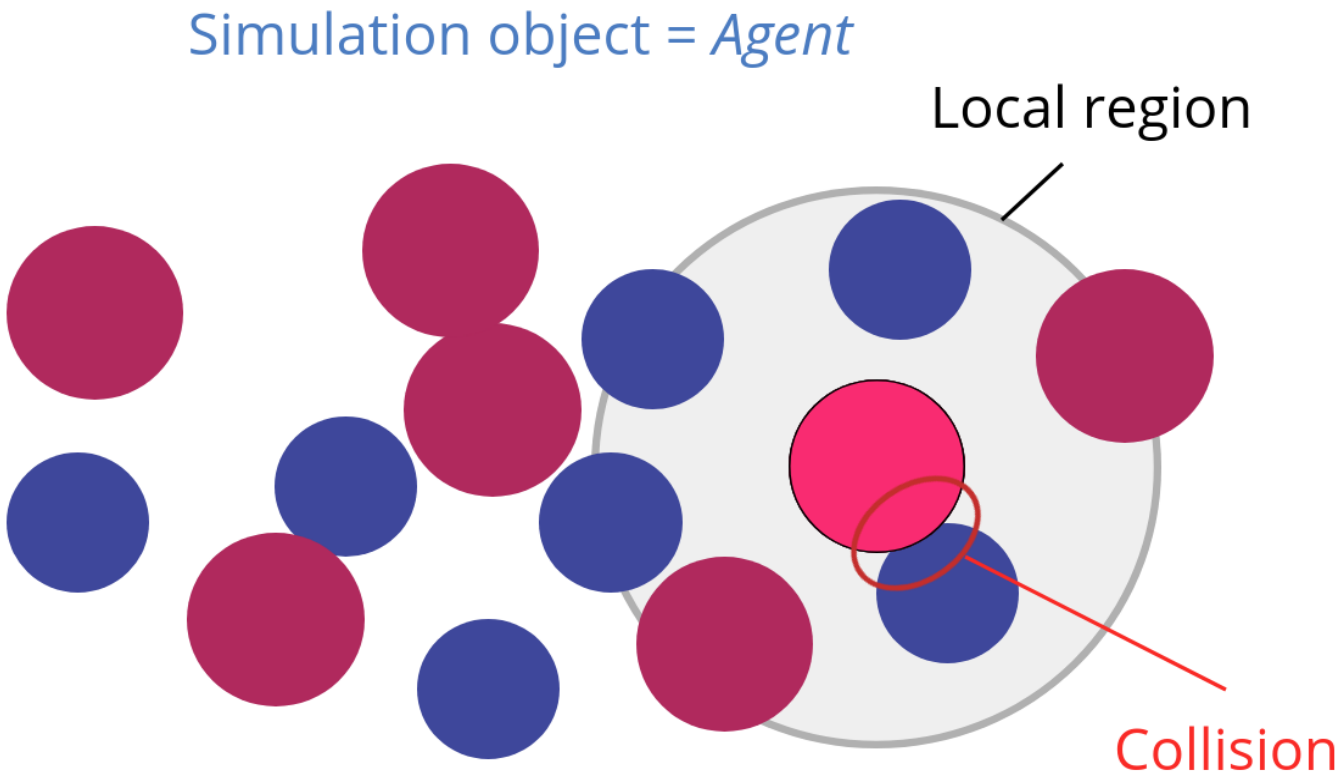


CERN openlab Technical Student

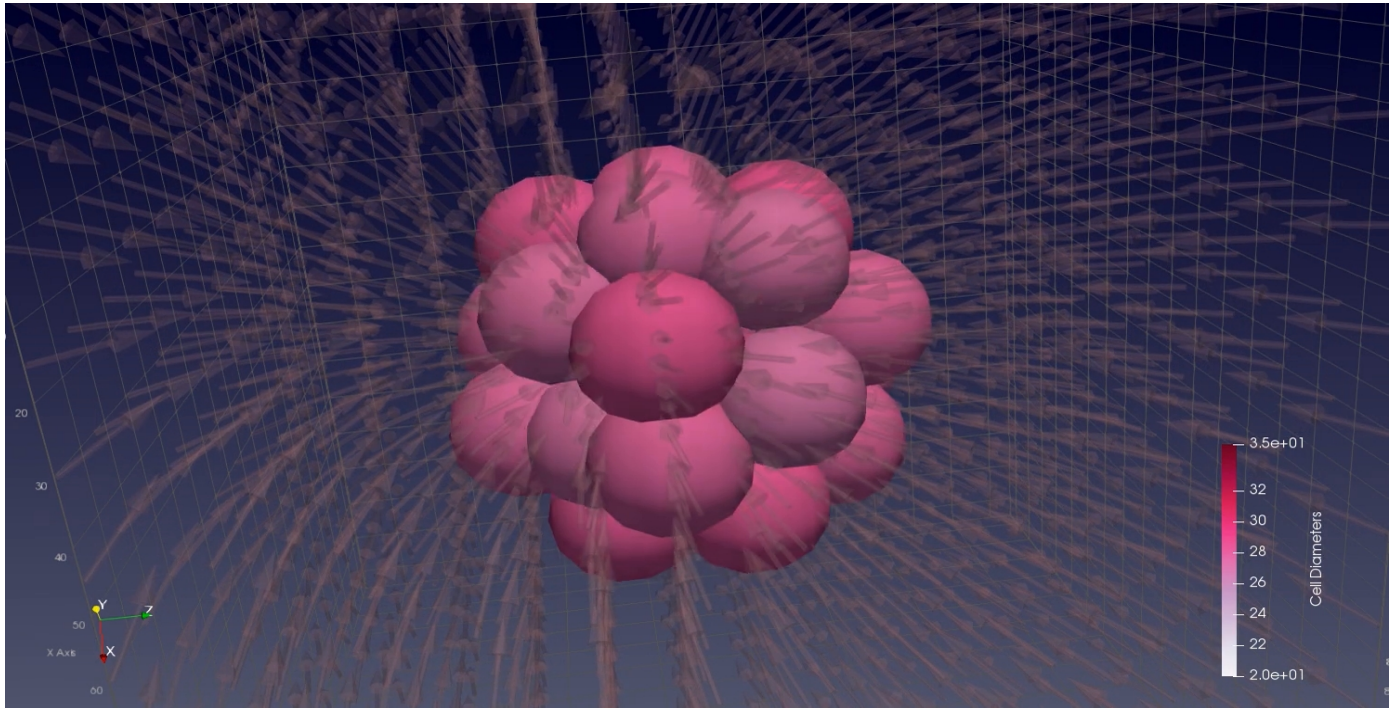
- Master eindproject bij CERN openlab
- Doorgaan met zomerstage project: BioDynaMo

Een “Knowledge Transfer” project: deel de kennis van CERN met andere wetenschappen

Agent-based simulation



Cellgroei



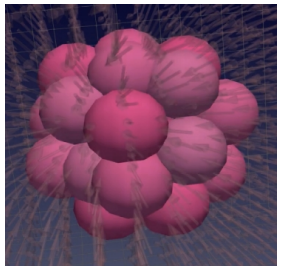
- Elke cell is geprogrammeerd
- Maak je eigen regels
- BioDynaMo zorgt ervoor dat het snel gesimuleerd wordt

Code

- Simulaties worden geschreven in C++ (zoals veel HEP software)
- C++ is snel, maar lastig voor beginners
- Natuurkundigen en bio-wetenschappers zijn er vaak niet bekend mee
- Hoe pakken we dit aan?

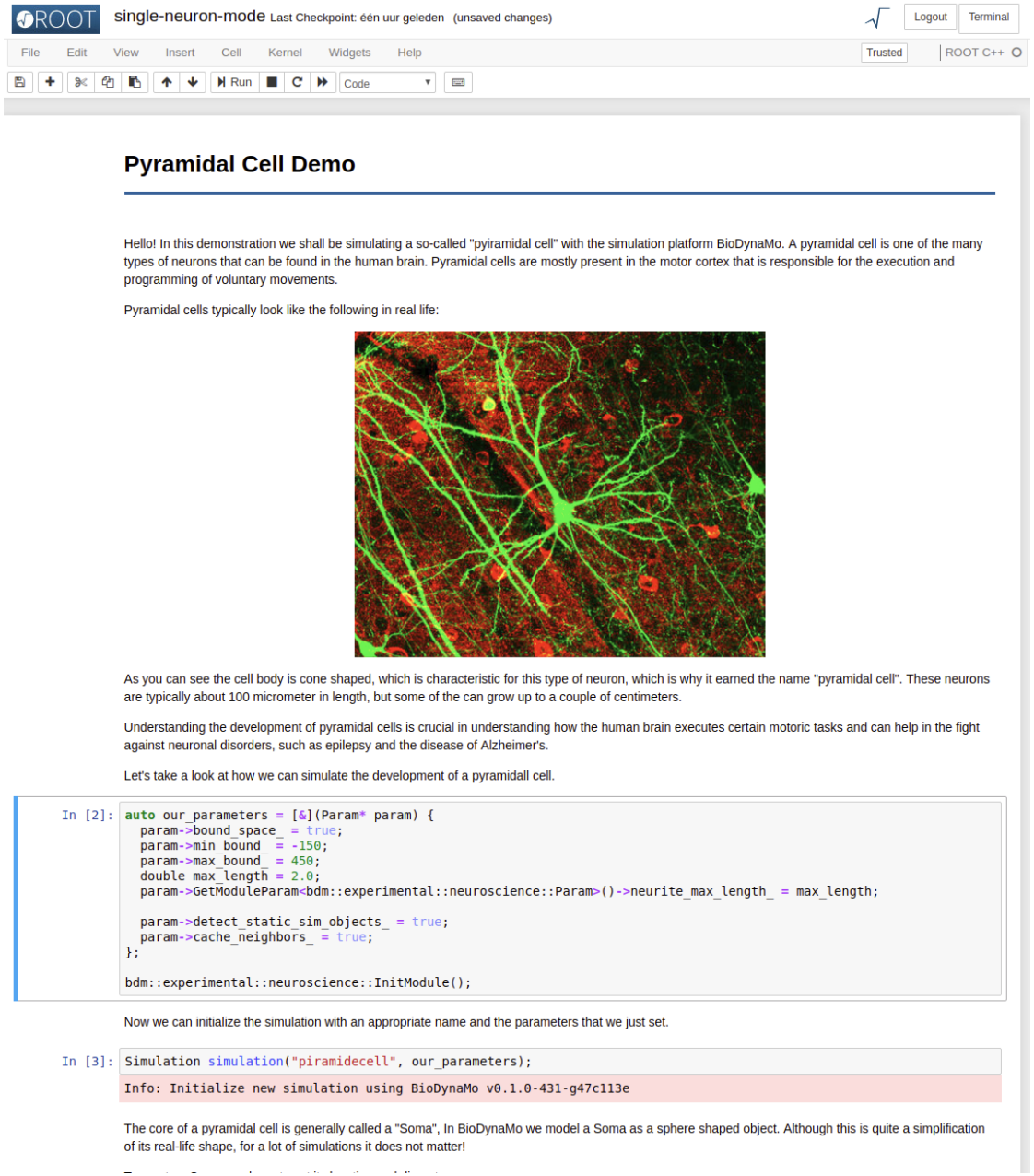
```cpp
#ifndef DEMO_CELL_DIVISION_MODULE_H_
#define DEMO_CELL_DIVISION_MODULE_H_

#include "biodynamo.h"

namespace bdm {

// ----------------------------------------------------------------------------
// This model creates a grid of 128x128x128 cells. Each cell grows untill a
// specific volume, after which it proliferates (i.e. divides).
// ----------------------------------------------------------------------------

inline int Simulate(int argc, const char** argv) {
  // Create new simulation
  Simulation simulation(argc, argv);
  // Since sim objects in this simulation won't modify neighbors, we can
  // safely disable neighbor guards to improve performance.
  simulation.GetExecutionContext()->DisableNeighborGuard();

  // Define initial model - in this example: 3D grid of cells
  size_t cells_per_dim = 128;
  auto construct = [](const std::array<double, 3>& position) {
    Cell* cell = new Cell(position);
    cell->SetDiameter(30);
    cell->SetAdherence(0.4);
    cell->SetMass(1.0);
    cell->AddBiologyModule(new GrowDivide());
    return cell;
  };
  ModelInitializer::Grid3D(cells_per_dim, 20, construct);

  // Run simulation for one timestep
  simulation.GetScheduler()->Simulate(1);

  std::cout << "Simulation completed successfully!\n";
  return 0;
}

}  // namespace bdm

#endif  // DEMO_CELL_DIVISION_MODULE_H_
```

Compileren en uitvoeren



Notebooks

- ROOT is een software ontwikkeld door CERN
- Ontwikkeld om physici te helpen bij analyzes
- ROOT Notebooks is een van de aspecten die het makkelijker maakt om analyze te doen
- Wij gebruiken het om biomedische wetenschappers te helpen met hun simulaties

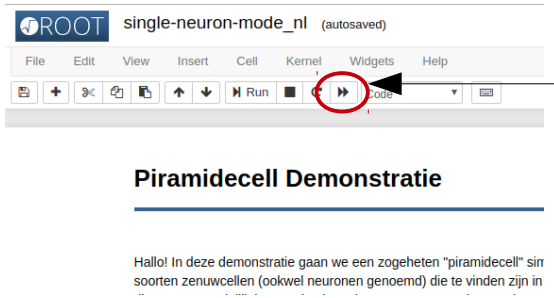


Demo tijd: simuleer je eigen piramidecel

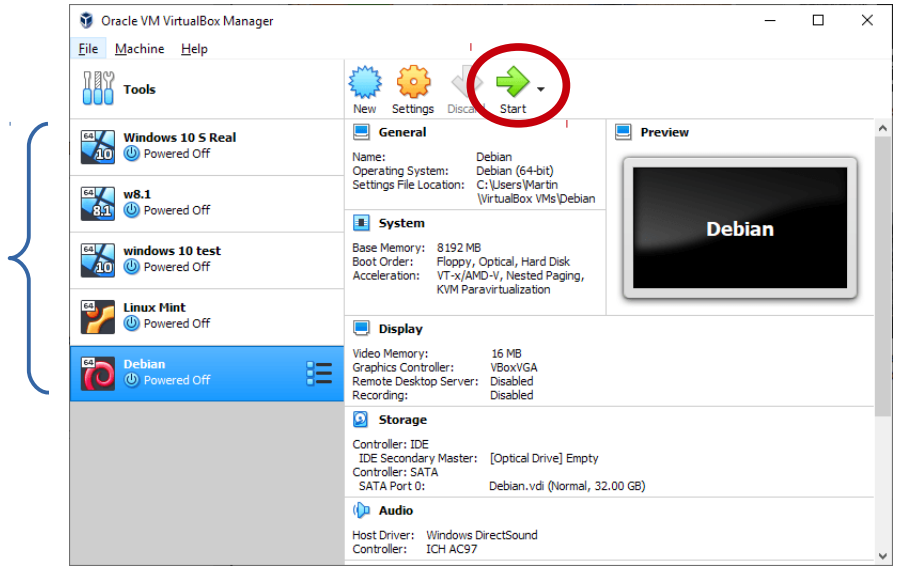
1) Open je laptop en login: wachtwoord is **Acoolpassword!**
2) Dubbelklik op het Oracle VirtualBox icoon
3) Selecteer **Ubuntu_BioMasterClass** in de lijst
4) Klik op **Start**
5) Neem de notebook door en voor de code uit door op Shift + Enter te klikken

   Probeer te begrijpen wat de code doet
6) Na het eenmaal doorgenomen te hebben: probeer wat parameters aan te passen en kijk wat het effect is.

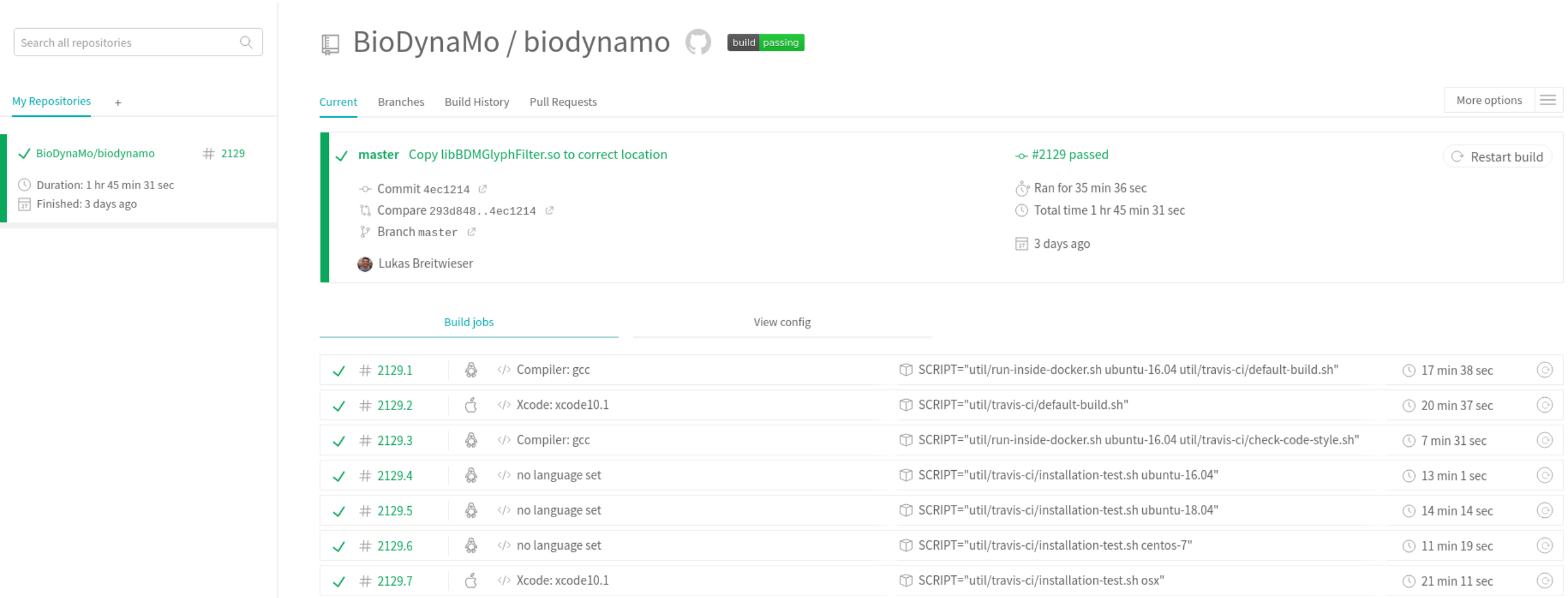
   Wie kan een tweede piramidecel maken?



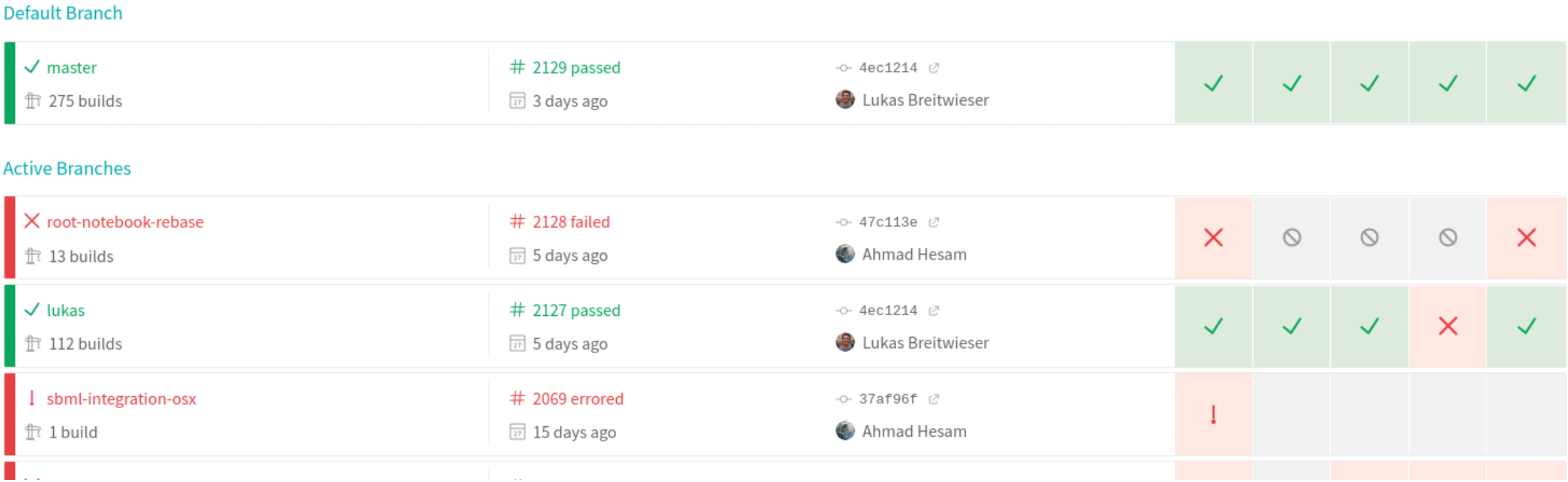
Klik om te resetten en alle cellen opnieuw uit te voeren



Continuous Integration



Continuous Integration



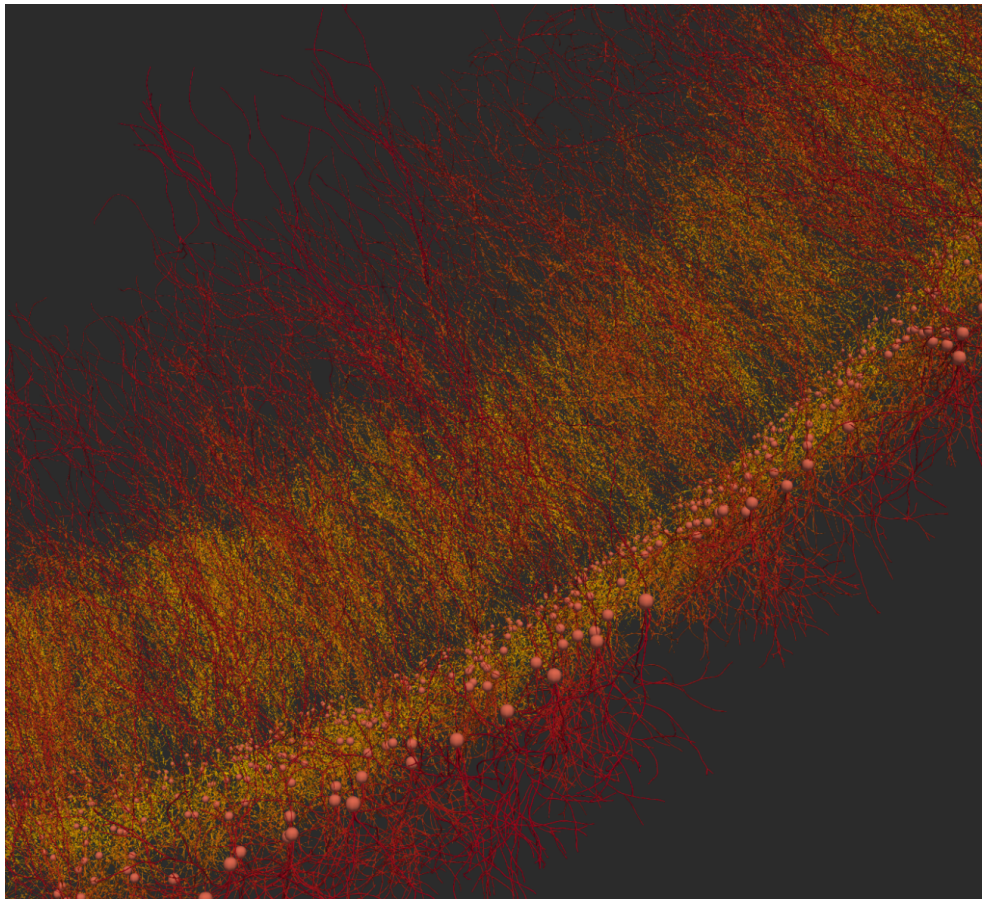
Open-Source Code

https://github.com/BioDynaMo/biodynamo

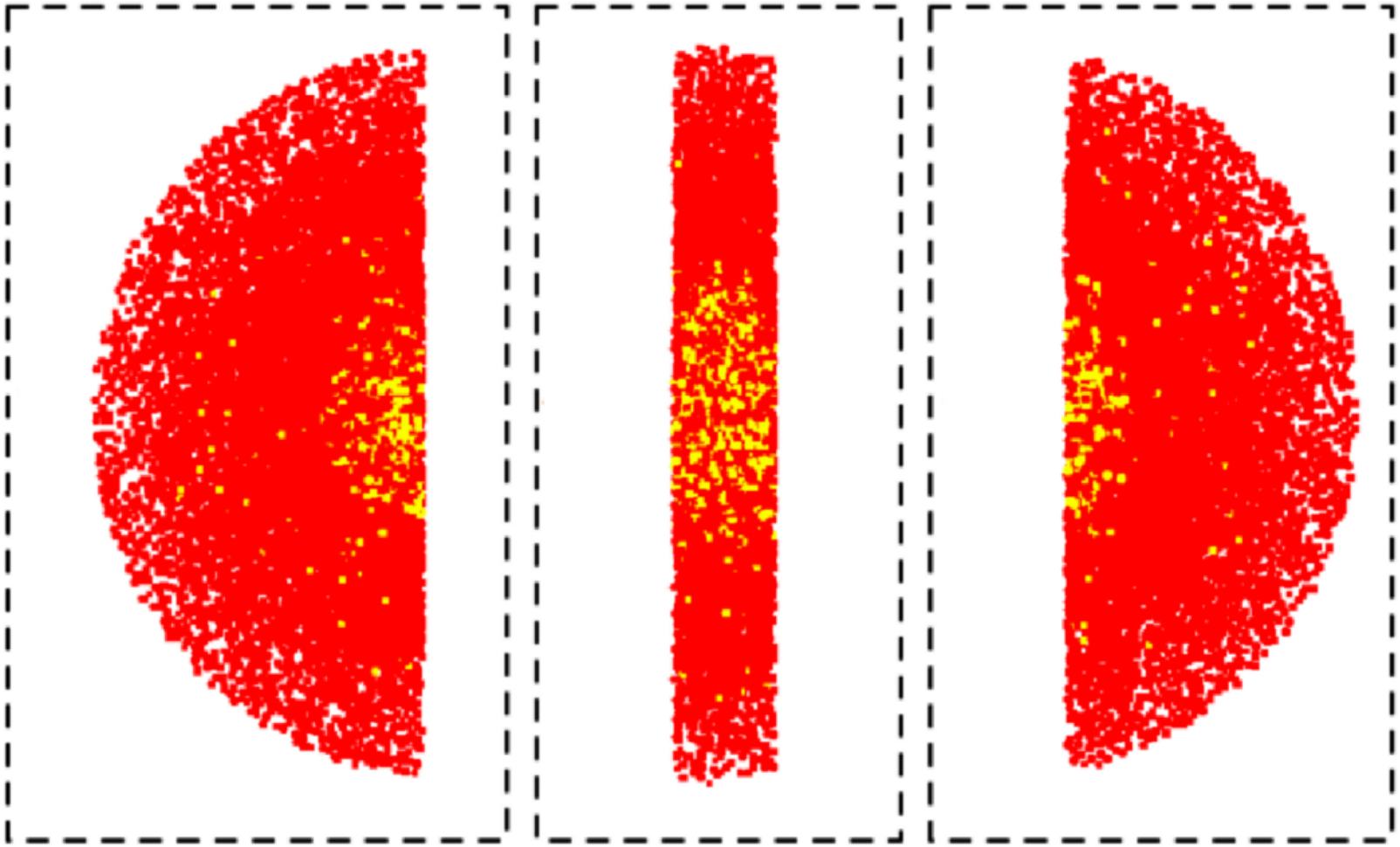
- Code is publiekelijk beschikbaar
- We moedigen contributies aan
- Het uiteindelijke doel is om BioDynaMo de defacto standaard te maken in biomedische simulaties

High-Performance Computing

- Simulaties van een paar neuronen kan gemakkelijk op een laptop
- Maar simulaties van duizenden neuronen moet gedaan worden op een krachtigere computer
- Sommige simulaties kunnen zo complex worden dat zelfs krachtige computers het niet aankunnen
- Oplossing: High-Performance Computing
  - Gedistribueerd: verdeel de simulatie over meerdere computers die tegelijkertijd werken
  - Hardware accelerators: GPUs / FPGAs



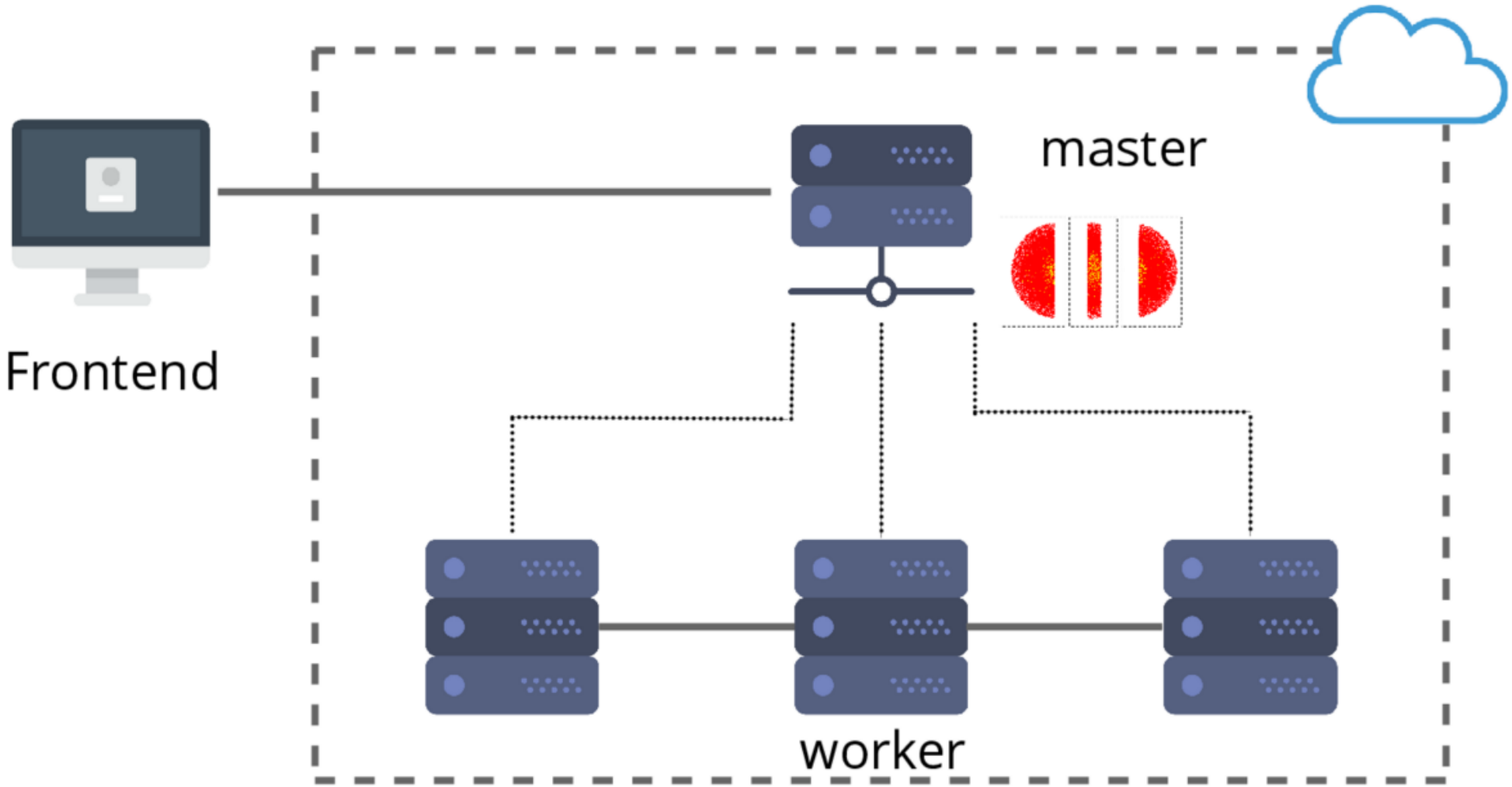
Domain-Decomposition



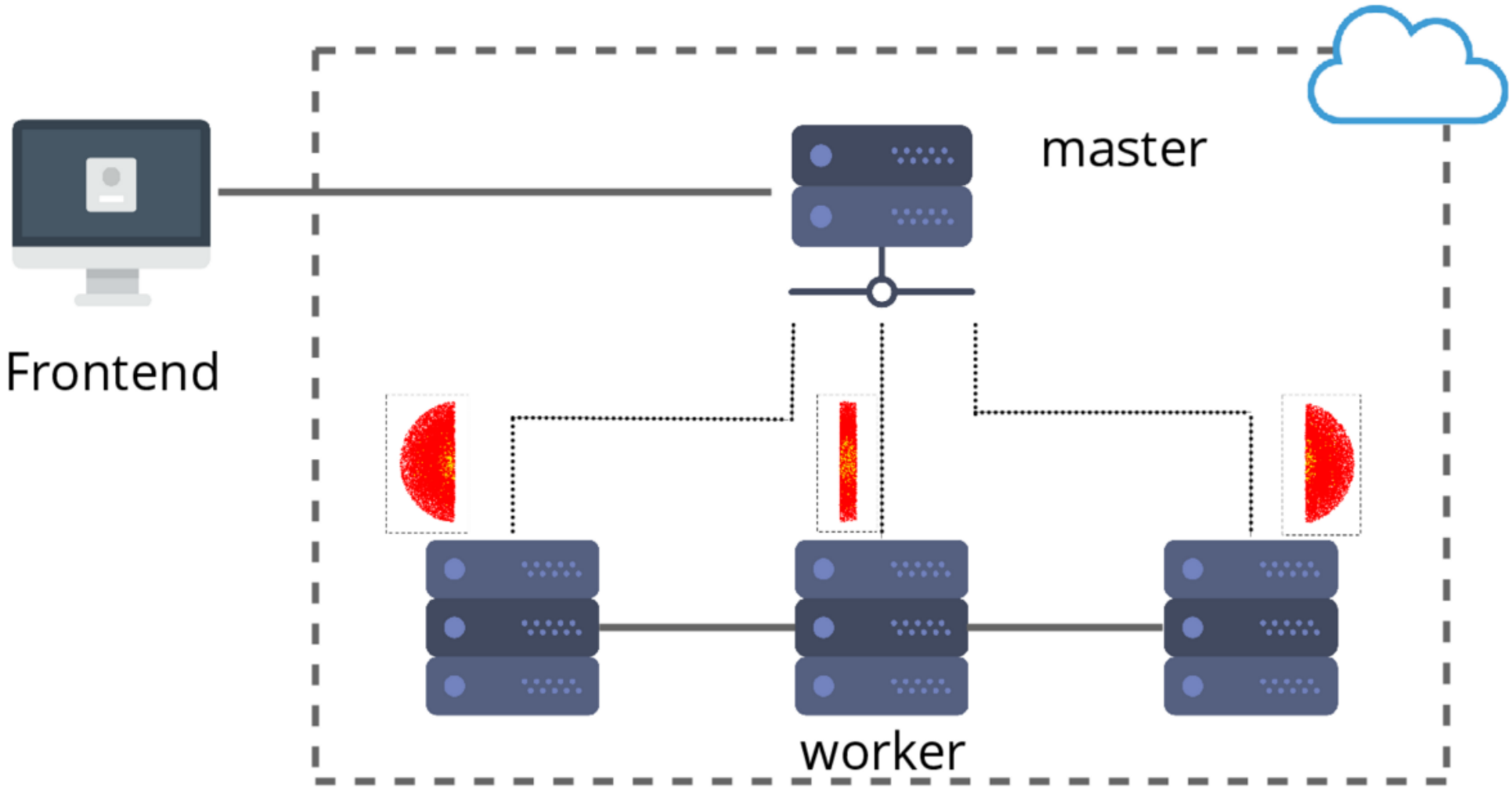
CERN openlab

Hauri, Andreas. *Self-construction in the context of cortical growth* . Diss. 2013.

Distributed Runtime



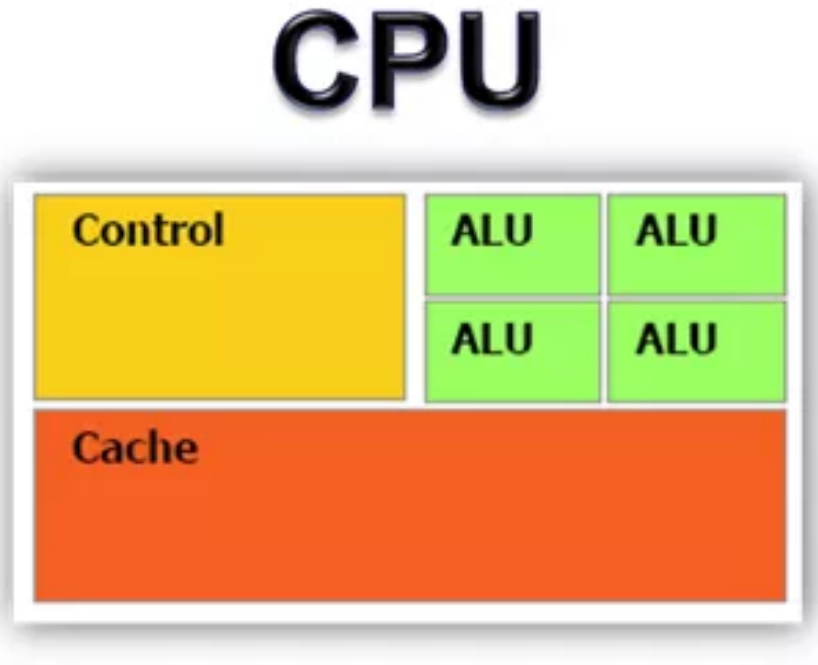
Distributed Runtime

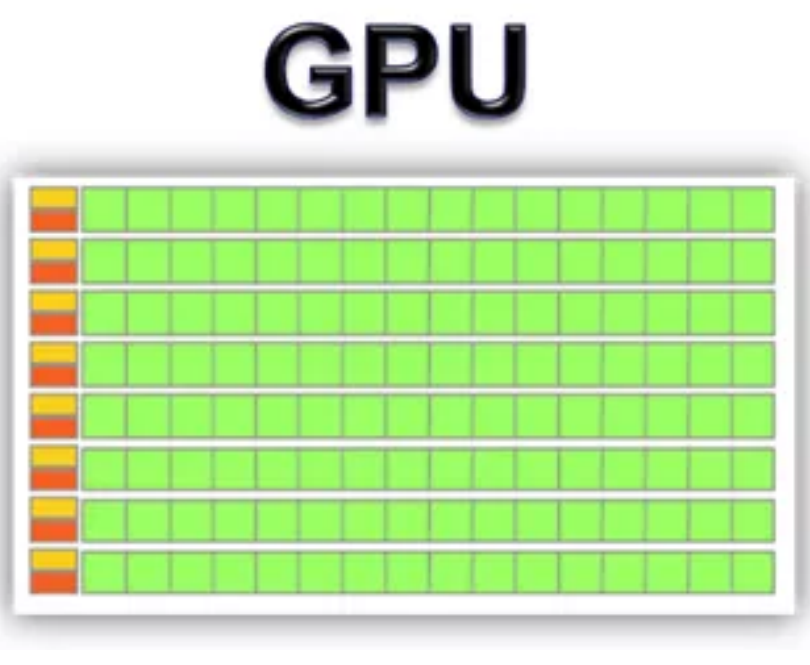


CPU vs GPU: A Demo

https://www.youtube.com/watch?v=-P28LKWTzrI

High-Performance Computing

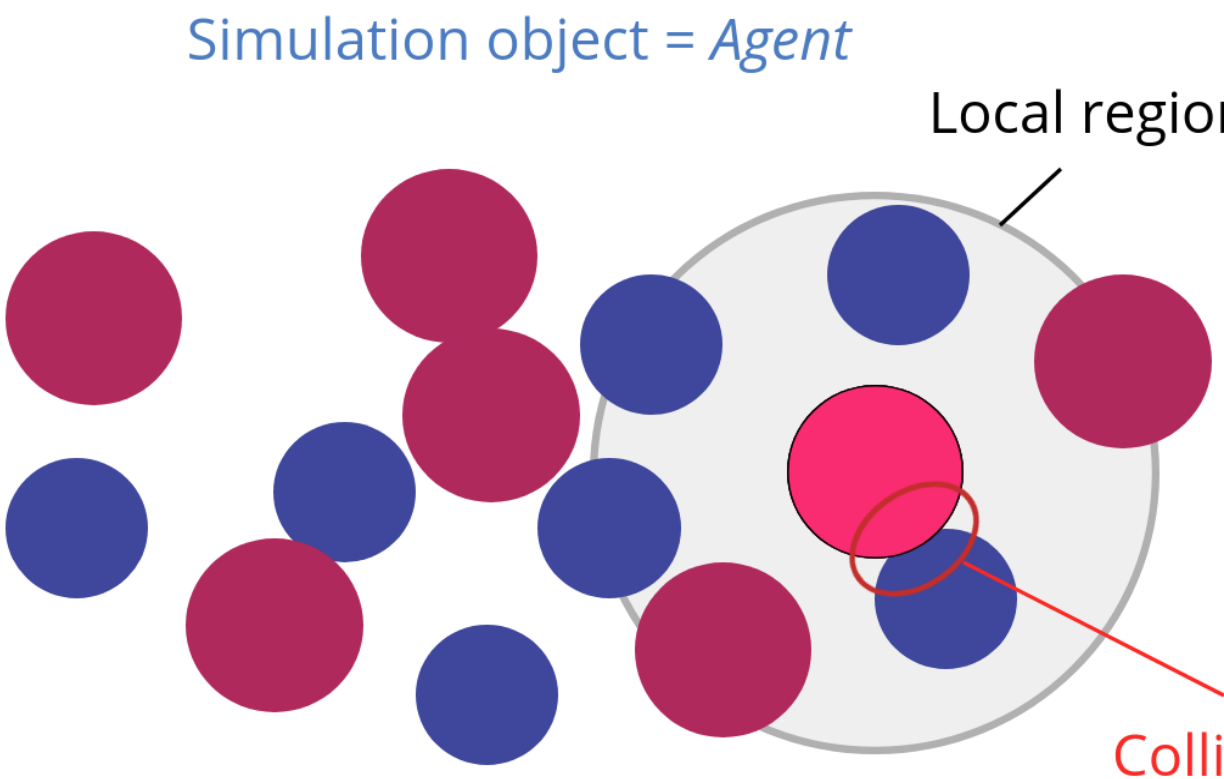




Een paar cores...

Een paar duizend cores!

Agent-based simulation



Mijn onderzoeksvraag

Kunnen we dit sneller doen?

≈ 80% van de tijd!

Resultaten

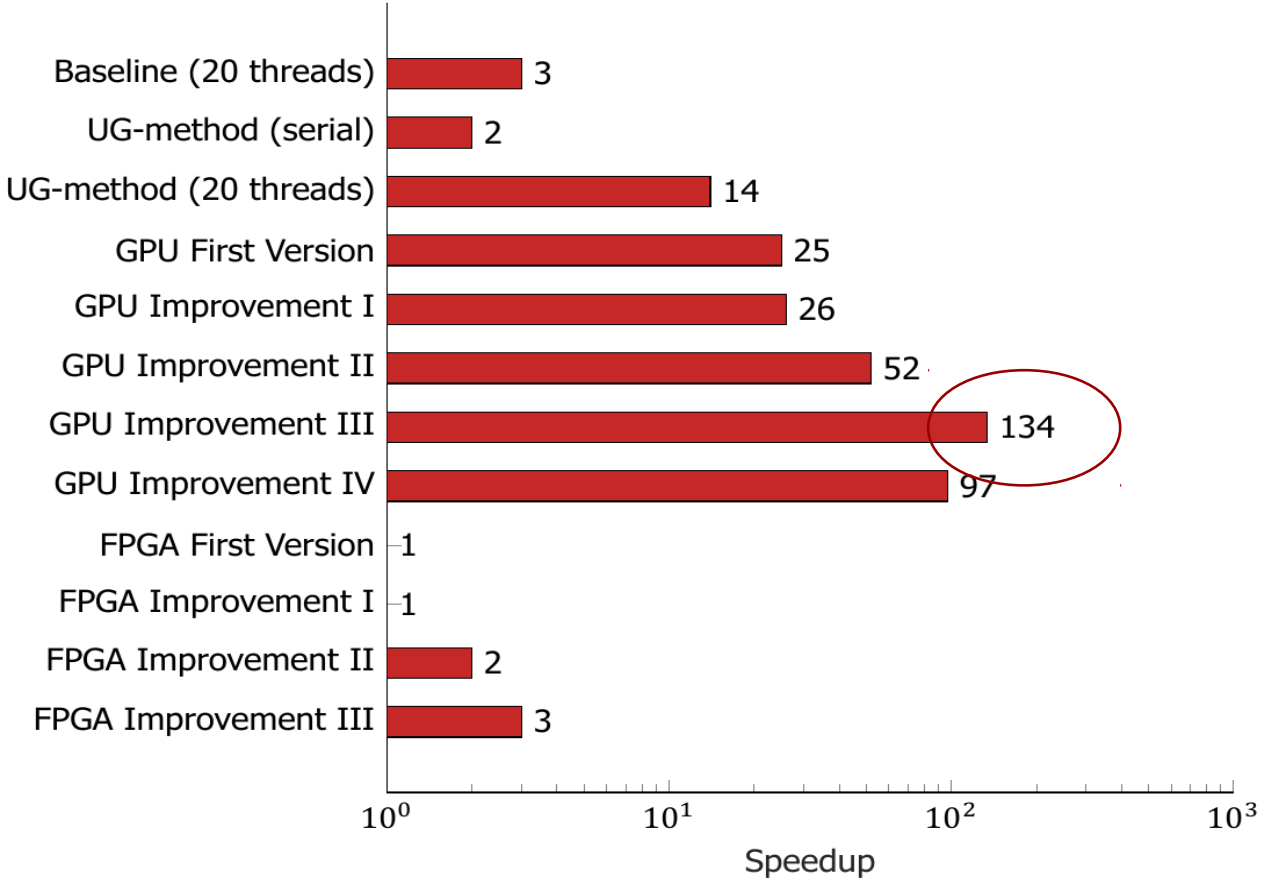


Figure 5.2: The speedup per simulation step with respect to the serial baseline version for all mechanical interaction implementations, with $num_{objects} = 2097152$.

QUESTIONS?

Ahmad Hesam

ahmad.hesam@cern.ch