

# **Parallel Programming Tools for Exploring Immune System Development**

OLIVER BINNS

## Overview

- Biological Background
- Motivation
- Solutions
- Results

PPSim is an existing simulation that was created to explore the development of clusters of lymphoid cells in the gut.

Aim to make it run faster using parallelism

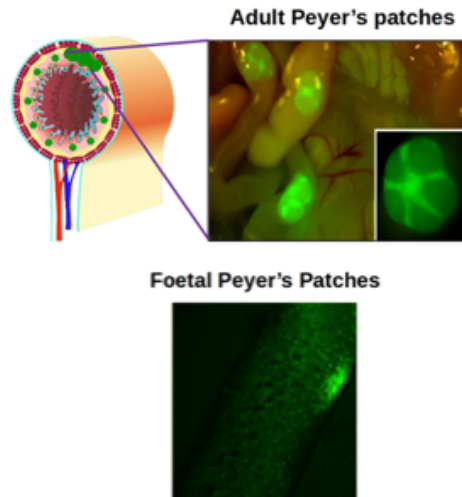
... 3 months to run

Down to 5.4 hours

In depth programming project

## What is it about?

- *PPSim*
  - ABM simulation
  - 94.265s per execution
  - 585,000 executions required



PPSim is an existing simulation that was created to explore the development of clusters of lymphoid cells in the gut.

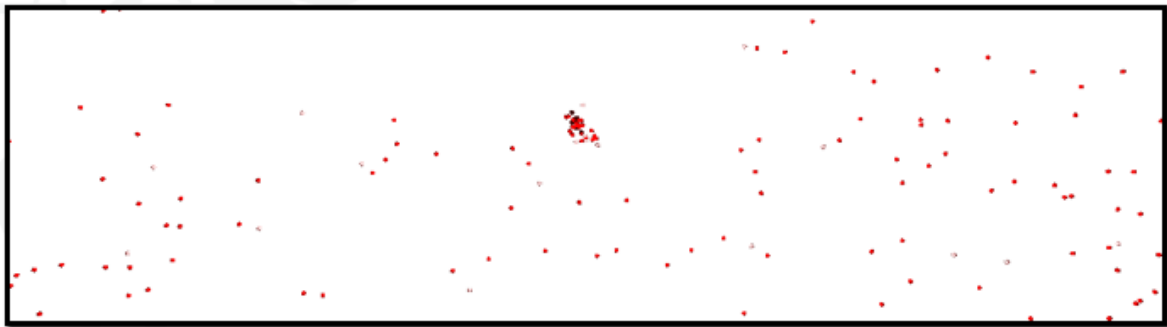
This is an inconvenient amount of time, even when run on a HPC.

Meant my supervisor Kieran had to wait 3 months for his results.. Play video games?

Get to the end and find a bug?

- Run again?

## Visualisation



Three types of cell- (agents)

Static LTo

LTi + LTin begin migration in

LTin activates LTo

LTo emits chemokines- a chemical which the LTi responds to

LTi begin to move towards LTo

Patch is formed!

## Motivation

- Short Term
  - Simulations used for novel biological findings
- Long Term
  - \$2.5bn R&D cost per drug \*
  - 3Rs: Animal Testing



\*: J. DiMasi et al, 2016

Obviously there's a lot of scope within biology for simulations to help with testing.

So far it's been used to gain a greater understanding of Peyer's Patch formation.

Further understanding means we could trigger a faster immune response to pathogens

Impact of work has gone to three pharmaceutical firms and one cosmetic firm

## Problems?

- Too Slow!
- Shortage of Computer Scientists
  - Efficient Parallel Programming is **HARD**.

PPSim is an existing simulation that was created to explore the development of clusters of lymphoid cells in the gut.

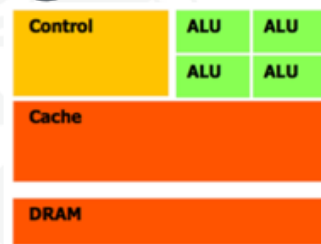
Aim to make it run faster using parallelism

... 3 months to run

Down to 5.4 hours

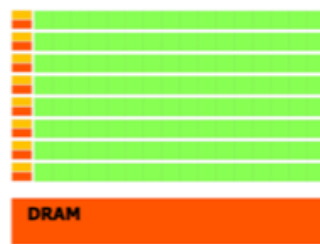
In depth programming project

# Program Parallelisation



CPU

4+ Cores



GPU

2500+ Cores

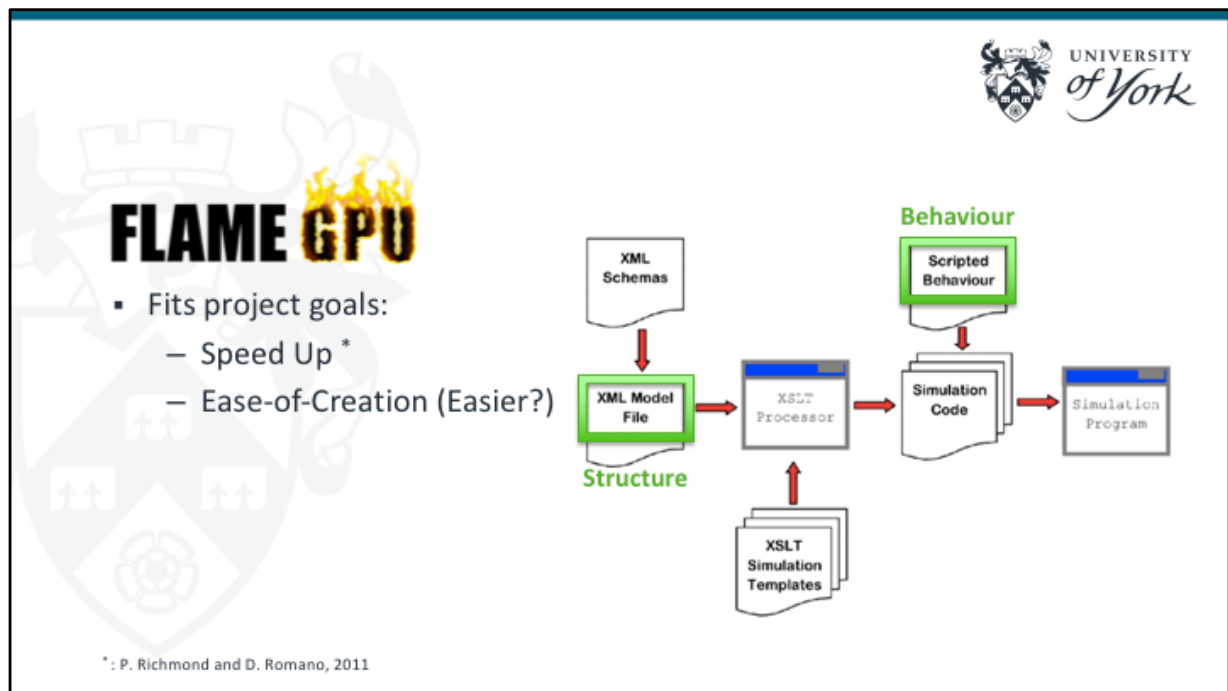
Agent-Based Modelling lends well to parallelism because each agent runs autonomously \*except for\* interaction.

Central Processing Unit: *Instruction Parallel*

Graphic Processing Unit: *Highly Data Parallel*

CPU can handle different tasks at once

GPU can go through sheer quantity of numbers, particularly in matrix form, performing the same operation



Analysed a number of ABM frameworks

Speed Up:

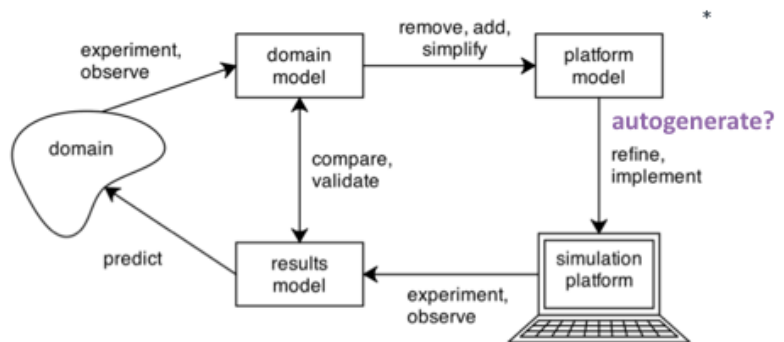
- Has shown performance on Desktop, comparable to HPC *like* that used for PPSim

Ease of Creation: FLAME GPU ***manages cross-thread communication***

- Still requires a computer scientist
- Model-Driven?
- Can go further



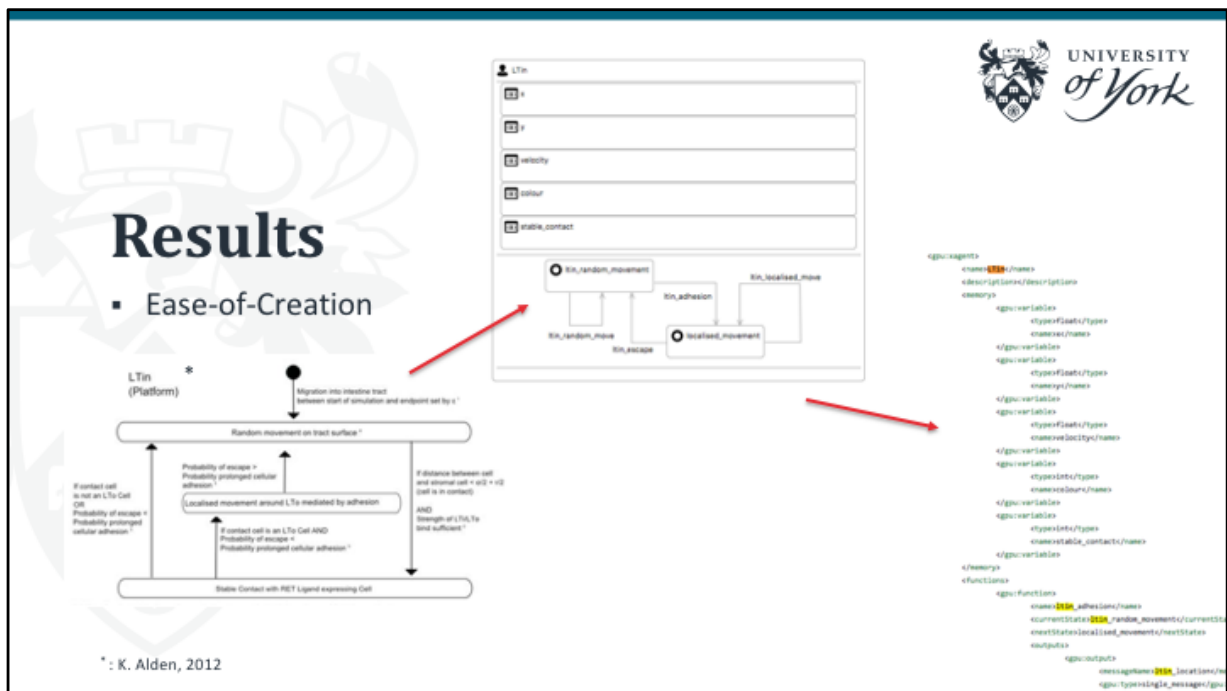
# Modelling Process



\*: M. Read, 2011

At the start of the project, we already *HAD* Domain + Platform Models  
Talk through CoSMoS

This project I attempted to begin development on tools which autogenerate the simulation platform



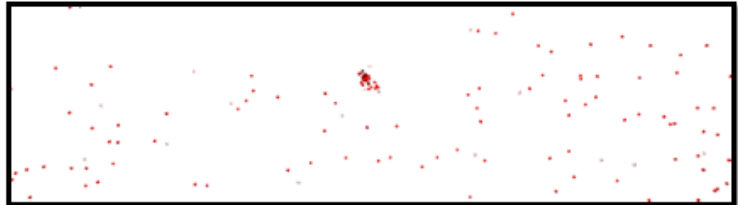
*For Domain Users: due to a shortage of Computer Scientists  
Epsilon for Eclipse, transform platform model into simulation model  
Software Engineer still needed for programming behavior script.*

*FLAME GPU –enhancements*

*Visited Sheffield for 2 hour meeting with Paul Richmond, May 1<sup>st</sup>*

## Results

- Speed Up
  - 94.265s vs 25.039s
- But not comparable:
  - Hardware Differences
  - Significant Implementation Differences
- Full Biological Analysis Required (Ongoing)



*Initially planned to compare speed up against PPSim*

*Face Validity*

- *Expected behaviour is present*
- *Hard to say more than that without full biological testing*

## Further Work

- PPSim v2
- FLAME GPU
- Software Generalisability
- Hardware Availability

*Full Statistical Analysis to show it really demonstrates Peyer's Patch development*

*Enhancements to FLAME GPU: already met Paul Richmond to discuss (1 May)*

*See if we can also generalise behaviour and allow this to be extracted from Domain Models*

*Extract implementation details, such as variable types*

*Lack of GPU availability:*

*Evaluate cross-platform GPU support, currently FLAME only supports NVIDIA*

# Any Questions?

All project code, report and slides available at:

[GITHUB.COM/OLIVER-BINNS/PRIY.GIT](https://github.com/Oliver-Binns/priy.git)