The University of York Department of Computer Science

Submitted in part fulfilment for the degree of MEng.

Parallel Programming Tools for Exploring Immune System Development

Oliver Binns

1st February 2018

Supervisor: Dr. Fiona Polack, Dr. Kieran Alden

Number of words = -1, as counted by `wc -w`.
This includes the body of the report only.

Abstract

More powerful computers are paving the way for realistic simulations of previously underexplored biological systems. As advancements in computing tend increasingly towards parallelism and distributed systems, these ever more complex simulations must take full advantage of this change in order to be computed in a reasonable time. In this project, we will explore some of the current technologies for doing so and propose new tools for the future.

Contents

List of Figures

List of Tables

List of Listings

1 Introduction

1.1 Project Overview

TALK ABOUT MODELING AND THE DESIRE FOR
Faster AND More Accessible Modelling

1.2 Motivation

1.3 Project Aims

In summary, the aims of this project are to

1. Aim 1
2. Develop a parallel implementation of an existing Peyer's Patch simulation and explore any speed increases that can be produced using General Purpose GPU programming.
3. Establish a firm grounding for the future development of new tools to allow fast, parallel simulations of biological systems to be easily created by non-technical users.
    a) Explore the findings of the new implementation and discuss how these can be generalised to new simulations.
    b) Discuss techniques for allowing non-technical users to easily create formal models that can be transformed into new simulation implementations.

1.4 Statement of Ethics

This project was conducted in accordance with the University of York's code of practice on ethics. This project does not involve human participants, so guidelines on informed consent and confidentiality will be met. No confidential medical data or personal information has been used

during the course of the project development. This project has involved no animal participation.

[Where did the data come from?!] The simulation of the biological model is for the purpose of developing understanding of applying GP-GPU methods to an agent based model of a biological system. It will not be used its current form to publish novel biological findings and does not fully simulate a biological process.

1.5 Report Structure

This report details the work done throughout the project and

Chapter 2 gives a general overview of simulations and the benefits and limitations of their use particularly with regard to computational biology.

Chapter 3 explores the limitations of simulations in more detail and proposes future solutions for these.

Chapter 4 details the development of an improved, inherently parallel, implementation of PPSim.

2 Background

2.1 Simulation

Computer simulations are used in a wide range of disciplines on applications such as video games, medicine, product development and even nuclear weapons.

2.1.1 Benefits

Feasibility

Exploring computer simulations is often far more feasible than exploring a real world environment. Video games allow players to experience scenarios that they may not otherwise get the opportunity to encounter. For example, car racing games are significantly cheaper and safer than real life racing. Within scientific discovery, simulations may be used as an alternative to real-world testing or to complement it.

Often real-world scientific testing may not be feasible for a number of reasons. Simulating the aerodynamics of a car virtually is far quicker and cheaper than creating multiple different prototypes for physical tests. Morality may be a factor, animal testing for cosmetic products or medicine is a good example of this. With nuclear weapons, legality is a key issue, as weapon testing is banned under a number of global treaties. Finally, these real-world tests may be too dangerous to perform, such as in the case of invasive medical examinations.

Accurate Results

2.1.2 Limitations and Constraints

While simulations seem very useful across a wide number of fields, there are some significant limitations as to where and how they can be used.

Insufficient Domain Knowledge

A simulation is based on a model of a system. A model is an abstraction from reality representing only the necessary key characteristics and behaviours of the system. A lack of knowledge regarding the domain of the simulation is one of the most significant constraints regarding its implementation. If this is the case, the model produced may be incorrect or abstractions may remove necessary detail required for the system to function as expected.

For complex systems, having too many abstractions from the original domain may invalidate the model and produce incorect results[CITATION NEEDED].

Compute Power

Complex models with too few abstractions from their domain may require significant computing power to simulate. Additional abstractions may not be possible as they may invalidate the model. In this case, cutting edge hardware will be needed for the simulation to be run in an acceptable time. Powerful hardware is expensive to access, so this may be a significant constraint on the ability to simulate.

Bugs

As with any form of computer program, mistakes can be made causing bugs to be present in the simulation code. Bugs may cause the simulation to be incorrect meaning any hypotheses and results are based on incorrect data.

This is linked to, but not the same as, having insufficent domain knowledge (as discussed above). Both of these limitations will cause the simulations fail silently, produce incorrect results with no immediately obvious failure[CITE]. However, while these problems are specific to simulation, they are not dissimilar from the issues that can occur from poorly designed real-world testing.

If the simulation needs to be safety-critical, developing it using formal methods and refinement may be a good way to ensure that no bugs are present in the code.

Skills Shortage

Advanced hardware alone will not necessarily allow a simulation to compute in a reasonable amount of time. Often, and particularly with the increasingly parallel modern architectures, the simulation code must be tailored to take advantage of the computing power available. Significant computer science skills shortages, across the world, significantly limit the possibility for cross-disciplinary work to utilise fast advanced simulations.

2.1.3 Agent Based Modelling

2.2 Biological Simulations

required as an alternative to invasive medical testing/animal testing

Simulations have even been proposed as a method for exploring a potential set of first principles and mathematics that are specific to biology which could even constitute a new subject- theoretical biology[1].

2.2.1 Existing Simulations

2.2.2 Cell Dynamics

2.2.3 PPSim

This project focuses on simulation as a tool for exploring biological systems at cell level. It uses the existing simulation of Peyer's Patch[2] and attempts to use parallel computer architectures in order to speed this simulation up.

Finally, I will propose a new tool, which builds on existing work in order to make this power available to non-technical users? or will I, probably not enough time for this!

3 Improving Simulations

As these simulations becoming more and more prevalent: tools need developing to allow less technical users to create these simulations easier and run faster...

3.1 Ease of Creation

Ease of creation (and maintenance) is an important feature for future simulation particularly due to the aforementioned computer science skills shortage (Section 2.1.2).

3.1.1 Flexible Modelling

Flexible Modelling tools could be a good method for allowing new simulations to be created more easily. Using flexible modelling, non-technical users would be able to create sketches which can be automatically processed by tools into formal models and prototype metamodels[3]. FlexiSketch is a good example of this and provides a good tool for creating models and metamodels for software development[4].

3.2 Speed Up

3.2.1 Machine Learning

A solution to the speed problem that has been proposed recently is to use machine learning on a small set of results to produce This has problems in that... likely affected by Standard Machine Learning issues? Bias? Overfitting?

3.2.2 Parallelism

Parallelism fundamentally changes the game and allows computers to keep following Moore's law even has engineers are struggling to make transisters ever smaller and smaller.[citation needed]

As modern computers tend further towards parallelism to keep providing the speed-ups that have been inherent in the industry over recent years, new parallel algorithms need developing in order to take full advantage of the computing power available.

CPU vs GPU

We are building on a previous project[5] which layed the groudwork for this. This previous project outlined the choice between CPU and GPU parallelism and makes the case for exploring GPUs- simplisitically put, this is due to the significantly greater speed ups that can be achieved.

OpenGL vs CUDA

A previous project compared OpenGL and CUDA

Agent Based Modelling

Agent Based Modelling is particularly well suited to parallelism as each agent makes its own decisions. Communication between agents may be a problem here!

Mention Flame (traditional) is a start..

3.2.3 FlameGPU

4 Methods

4.1 The Domain Model

The domain model is taken from an existing simulation...

4.2 The Platform Model

OPTIONS:
Custom Code?[5] not well tested less easy to update to support new tools and hardware, new CUDA GPUs Each custom code simulation must be updated separately
model based -> FlameGPU[6] restricted to the framework limitations can cells migrate into the system?!

4.2.1 Testing

Talk about how the model was tested to ensure correctness
Mention missing link in (incorrect) model from Kieran's paper

5 Results and Evaluation

5.1 Findings

[7] could be important for evaluating the performance of FlameGPU against original PPSim

5.2 Conclusion

5.3 Further Work

5.3.1 Hardware Availability

5.3.2 Software Generalisibility

Bibliography

[1] D. Noble, 'The rise of computational biology', *Nature Reviews Molecular Cell Biology*, vol. 3, pp. 459–463, 2002. DOI: 10.1038/nrm810.

[2] K. J. Alden, 'Simulation and statistical techniques to explore lymphoid tissue organogenesis', PhD thesis, University of York, Heslington, York, 2012.

[3] R. F. Paige, A. Zolotas and D. Kolovos, 'The changing face of model-driven engineering', in *Present and Ulterior Software Engineering*, M. Mazzara and B. Meyer, Eds. Cham: Springer International Publishing, 2017, pp. 103–118. DOI: 10.1007/978-3-319-67425-4_7. [Online]. Available: https://doi.org/10.1007/978-3-319-67425-4_7.

[4] D. Wüest, N. Seyff and M. Glinz, 'Flexisketch: A lightweight sketching and metamodeling approach for end-users', *Software & Systems Modeling*, Sep. 2017, ISSN: 1619-1374. DOI: 10.1007/s10270-017-0623-8. [Online]. Available: https://doi.org/10.1007/s10270-017-0623-8.

[5] P. Drew, 'Parallel programming tools for exploring immune system development', Master's thesis, University of York, Heslington, York, 2017.

[6] P. Richmond, D. Walker, S. Coakley and D. Romano, 'High performance cellular level agent-based simulation with flame for the gpu', *Briefings in Bioinformatics*, vol. 11, no. 3, pp. 334–347, 2010. DOI: 10.1093/bib/bbp073.

[7] A. Arcuri and L. Briand, 'A practical guide for using statistical tests to assess randomized algorithms in software engineering', in *Proceedings of the 33rd International Conference on Software Engineering*, ser. ICSE '11, Waikiki, Honolulu, HI, USA: ACM, 2011, pp. 1–10. DOI: 10.1145/1985793.1985795. [Online]. Available: http://doi.acm.org/10.1145/1985793.1985795.

6 Appendix

6.1 Simulation Parameters

6.2 Cell Data Structures