Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

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

Simulating heterogeneous behaviours in complex systems on GPUs

Mozhgan K. Chimeh*, Paul Richmond

University of Sheffield, United Kingdom

ARTICLE INFO

Article history:
Available online 10 February 2018

Keywords:
Agent Based Modeling
GPGPU
Data parallel algorithms
Simulation
FLAME GPU

ABSTRACT

Agent Based Modelling (ABM) is an approach for modelling dynamic systems and studying complex and emergent behaviour. ABMs have been widely applied in diverse disciplines including biology, economics, and social sciences. The scalability of ABM simulations is typically limited due to the computationally expensive nature of simulating a large number of individuals. As such, large scale ABM simulations are excellent candidates to apply parallel computing approaches such as Graphics Processing Units (GPUs). In this paper, we present an extension to the FLAME GPU[1] [1] framework which addresses the divergence problem, i.e. the challenge of executing the behaviour of non-homogeneous individuals on vectorised GPU processors. We do this by describing a modelling methodology which exposes inherent parallelism within the model which is exploited by novel additions to the software permitting higher levels of concurrent simulation execution. Moreover, we demonstrate how this extension can be applied to realistic cellular level tissue model by benchmarking the model to demonstrate a measured speedup of over 4x.

1. Introduction

A complex system is a system which involves a large number of interacting entities/individual agents. Agent Based Modelling (ABM) is a method of understanding the behaviours of such system. It allows the behaviour of entities comprising a complex system, to be described as a set of individual behaviours or rules. In order to explore emergent properties resulting from such systems, the behaviours and interactions between entities can be simulated.

When simulating complex models using ABMs, increasing the size of the model or behavioural complexity of the individuals leads to an increase in the computational requirements. The use of parallel computing resources offers a viable solution to constrain these requirements and sustain reasonable simulation times even for very large or complex systems. Modern Graphics Processing Units (GPUs) contain hundreds of arithmetic processing units that can be utilised to achieve significant acceleration for computationally intensive scientific applications. GPUs allow a personal computer to be transformed into a personal supercomputer, providing up to 12 Trillion Floating Point Operations per Second (TFLOPS) in consumer hardware (NVIDIA TITAN Xp). Whilst computationally powerful, GPUs differ significantly in the hardware design of modern CPUs.

* Corresponding author.
E-mail addresses: m.kabiri-chimeh@sheffield.ac.uk (M.K. Chimeh), p.richmond@sheffield.com (P. Richmond).
URL: http://mkchimeh.staff.shef.ac.uk (M.K. Chimeh), http://paulrichmond.shef.ac.uk (P. Richmond)

[1] Flexible Large-scale Agent-based Modelling simulation platform on GPU

https://doi.org/10.1016/j.simpat.2018.02.002

Writing programs which are able to utilise the high levels of parallel performance requires considerable knowledge of data parallel algorithm design and extensive optimisation skills.

At first glances, ABMs appear well suited to a GPUs architecture. Each of the individuals (or agents) within the system can be simulated in parallel. Unfortunately, this is not the case, one has to consider methods of handling communication (and hence synchronisation) among individual agents, sparsity within populations as a result of agent creation and death (life and death), and dealing with heterogeneous behaviours within the group (divergence). Solving these challenges requires algorithms to be designed or applied to each of this issues which uses data-parallelism to scale to the GPUs hundreds of tightly coupled vector processing units. FLAME GPU is an extended version of the FLAME (Flexible Large-scale Agent-Based Modelling Environment) framework and is a mature and stable Agent-Based Modelling simulation platform that enables modellers from various disciplines like economics, biology and social sciences to easily write agent-based models, specially for GPUs. Developed since 2008, FLAME GPU has previously addressed the challenges of communication among agents and life and death of agents [2]. The software is able to provide agent based modellers with the ability to target readily available GPUs capable of simulating many millions of interacting agents. Previous work has demonstrated that performance can easily exceed that of traditional CPU based simulators [3]. One of the key design considerations is the use of a high level ABM syntax to abstract the complexities of the underlying GPU architecture away from modellers. This ensures that modellers can concentrate on writing models without the need to acquire specialist knowledge typically required to program GPU architectures.

Aside from relatively simple (or artificial benchmark) models, complex systems are typically heterogeneous with respect to the exhibited behaviour of individuals. In other words, agents within ABM rarely have identical sets of rules over their complete life-cycle. For instance, an agent representing a typical biological cell might have a number of distinct behavioural stages as it differentiates between cell types. The issue of divergence or divergent behaviour within populations of agents is largely unsolved by FLAME GPU. The FLAME GPU user guide actively encourages the use of large population sizes to ensure that the GPU device (which requires many parallel operations on agents) can be fully utilised when executing threads corresponding to agent behaviour. Typically populations of many thousands of agents are required to ensure good utilisation and as such agent behaviours often contain high levels of divergence.

The paper outlines best practices and considerations for model developers to systematically address agent heterogeneity on GPU architectures. The objective of this paper is therefore to propose a specific set of design considerations for implementation of agent models to ensure efficient execution on the GPU by reducing divergence. To evaluate the performance improvement, the design considerations have been tested with novel implementations (Section 5). The results show considerable improvements on the state-of-the-art, demonstrating the first formal provision of guidelines for best practice in addressing agent heterogeneity on GPU architectures in a consistent manner.

This paper aims to consider the problem of agent divergence offering the following novel contributions;

- We present an extension to FLAME GPU which exploits inherent parallelism within models to demonstrate efficient concurrent execution of divergent behaviours.
- We introduced key design considerations for modellers which encourage high levels of parallelism within a model whilst minimising the effect of divergent behaviour.
- We apply this extension to a heterogeneous cellular level biological model of tissue wound formation to demonstrate a simulation performance speedup of 4x on model sizes of up to 131k agents.

The rest of this paper is organised as follow: Section 3 describes the related issues and challenges of multi-agent simulation on GPUs. Section 4 presents design considerations required to implementing a heterogeneous model with high degrees of model parallelism. Section 5, reports the result of our experimental evaluation. Finally, we draw our conclusions in Section 7.

2. Background

Simulation is a tool for researchers to study and better understand the behaviour of a system, as well as predicting its performance. A number of different methods may be used to represent the system based on the characteristic of the model. Typically complex system can be represented as top-down by using sets of equations to model system level behaviour or bottom up by modelling the individuals with the system as agents. Simulation of the ABM, often referred to as Multi-Agent Simulation (MAS), is a method of studying complex systems [4] which provides a natural modelling approach as often the individual levels behaviour are well understood.

Due to the availability of increasing levels of computing power, a number of simulation frameworks have been developed and applied in different fields of engineering and science. Examples include Swarm [5], NetLogo [6], FLAME [7], Mason [8], and MCMAS [9]. D-MASON (Distributed Mason) is the distributed version of MASON MAS simulation framework developed to address the limitations on scalability (increasing the number of population) with respect to distributed hardware configurations. It uses remote agent invocation to communicate between agents to effectively address issues of synchronisation between many communicating CPU processors. FLAME uses a formal method of agent representation based on communicating X-Machines which solves synchronisation of communication by ensuring that only indirect communication through messages is permitted. The use of message boards ensures that communication can be limited to only agents where there