Applying Graphics Processing Unit Technologies to Agent-Based Simulation

Mitchell Welch
School of Science and Technology, University of New England, Australia

Paul Kwan
School of Science and Technology, University of New England, Australia

INTRODUCTION

In recent years, agent-based modelling has emerged as a successful approach for simulating complex systems across numerous domains, and for a wide range of purposes including social science, ecology, biology and epidemiology(Berger, 2001; Busing & Mailly, 2004; Connell, Dawson, & Skvortsov, 2009; D'Souza, Marino, & Kirschner, 2009; Elliston & Beare, 2006; Funk, Gerber, Lind, & Schillo, 1998; Minar, Burkhart, Langton, & Askenazi, 1996; Schelhorn, O'Sullivan, Haklay, & Thurstain-Goodwin, 1999). Agent-based models are developed around the principle of conceptually breaking complex systems down into individual components referred to as *Agents*. These components can theoretically represent any arbitrary level of detail within the system and have their own simulated behaviour and state. This system of interacting agents can be used to model complex phenomena from the bottom-up, allowing scientists to develop rich simulations capable of supporting experimentation at different conceptual levels within the system. The capability that this modelling approach provided has led to the use of Agent-Based Models (ABM) in a decision-support role for governments and industries, and fostered a demand for modelling systems with increasing levels of detail within the individual agents and their interactions, which in turn provides an even greater scope for experimentation. In addition to this, ABMs are being applied to larger scale systems, modelling more complex phenomena as data become more readily available at higher levels of detail.

The broadening in scope, coupled with the requirements for higher levels of detail, have driven an increase in the computational requirements of agent-based simulations, both in terms of memory and processor clock cycles. This is mainly due to the autonomous nature of the individual agents within the simulation, the state information they contain, and the interactions that occur between them. In simple terms, as the complexity of each individual agent's processing and/or the number of agents appearing in a simulation increases, the number of computational operations performed in each iteration of the simulation also increases. This challenge is further compounded by the requirement for interactions between the agents in order to simulate the phenomena. Likewise, a simulation's memory requirements also grow as the amount of state information stored within each agent and the number of agents increase.

Graphics Processing Unit(GPU) technologies, originally designed for carrying out the arithmetic processing required to produce complex three-dimensional graphics, have emerged as a high performing platform that is suitable for many general-purpose processing tasks. The development of technologies such as Nvidia's Compute Unified Device Architecture (CUDA)(NVIDIA, 2010) and OpenCL(AMD, 2010) provide the much needed frameworks for harnessing GPU's general-purpose processing capability, using a client-server architecture. The GPU programming approach has been successfully applied to large-scale agent-based modelling systems, resulting in promising performance gains and enhanced modelling capabilities. This article summarises the agent-based simulation approach, highlights the issues and challenges in applying GPU technologies to agent-based simulations, and reviews several real world implementations that have made use of the GPU technologies. The article concludes with a discussion of the performance gains

DOI: 10.4018/978-1-4666-5888-2.ch117

witnessed in the implementations reviewed, and briefly explains their approaches in dealing with the implementation challenges.

The rest of this article will be organised as follow. The second section provides a background of the agent-based modelling approach and an overview of current GPU technologies. The third section reviews a selection of recent real-world implementations of agent-based modelling systems that has utilised the GPU technologies. The final three sections discuss the potential performance gains in applying the GPU technologies, future developments, and present the concluding remarks for the article.

BACKGROUND

Agents and Agent-Based Simulation

Agent-based Modelling is connected to many fields and applications including complex systems science, computer science, and traditional modelling. It draws its conceptual origin from the Artificial Intelligence (AI) and *Complex Adaptive Systems (CAS)* research (Holland, 2006).There is no strict or universal definition for what actually constitutes an agent within an ABM. In some systems, any individually identifiable component is considered an agent (Macal & North, 2006). However, a far more common trend in literature such as found in specific simulations, CAS and AI systems(Russell & Norvig, 2003) is to define agents as identifiable discrete objects that are self-contained and autonomous, which are responsible for managing their own interactions, states and data. These agents can interact and update their states in discrete time steps within the simulation as well as perceive the environment in which they exist. Agents can be heterogeneous in nature within a simulation; a single model can contain limitless 'types' or classes of agents. The systems developed in (Grecu & Gonsalves, 2000; Higgins & Richardson, 1995; Luke, Cioffi-Revilla, Panait, & Sullivan, 2004; Minar, et al., 1996; Richmond, Coakley, & Romano, 2009b; Sonnessa, 2003) all specify multiple types of agents that interact to simulate their respective phenomena.

Agent-based simulation is a bottom-up approach for modelling phenomena. Agents, representing low-level components in a complex system, interact and produce output. These interactions determine the collective behaviour of the higher-level system. This property is referred to as *emergence* or *emergent behaviour* and can be illustrated by using an example of the human brain and its functions. High level ideas, states-of-mind and emotions are all determined by the different biochemical reactions and millions of cellular interactions that take place within the brain. Small changes within the biochemical interactions can combine to produce higher-level changes in state-of-mind and emotion. These chemical interactions are analogous to the agents in a complex system. Emergent traits within a complex system can be broadly characterised as *weak* or *strong*. Strongly emergent traits are those that cannot be directly traced back to one of the system components, but arise as a result of the interactions between the components. This is in contrast to weakly emergent traits, which are directly traceable to components within the complex system(Bedau, 1997).

This bottom-up approach, along with the autonomous, intelligent nature of the simulation's individual agents, introduces significant processing overhead into the simulation. In simple terms, as the complexity of the individual agent's processing and/or the number of agents in a simulation increases, the number of computational operations performed in each iteration of the simulation increases. A simulation's memory requirements increase as the amount of state information to be stored within each agent and the number of agents increases.

GPU Technologies for General Purpose Computation

Graphics Processing Unit (GPU) technology, driven by the increasingly lucrative computer game industry, has developed rapidly in the last decade, and is now providing the raw computational power to perform the large quantities of floating-point operations required to produce complex three dimensional graphics. A modern GPU is developed around the *many-core* processing paradigm, where large numbers of specialised processor cores are used to achieve a high throughput of operations in parallel(Kirk & Hwu, 2010). There is a significant difference between this design and that of a standard desktop/notebook computer's central processing unit (CPU). CPUs use a smaller number of cores (usually 2 to 8 in current systems) that are designed with high clock speeds and large quantities of on-chip memory