FLAME: Simulating Large Populations of Agents on Parallel Hardware Architectures

Mariam Kiran[*]
Department of Computer Science
University of Sheffield
Sheffield, UK
m.kiran@dcs.shef.ac.uk

Paul Richmond[†]
Department of Computer Science
University of Sheffield
Sheffield, UK
p.richmond@dcs.shef.ac.uk

Mike Holcombe
Department of Computer Science
University of Sheffield
Sheffield, UK
m.holcombe@dcs.shef.ac.uk

Lee Shawn Chin
Software Engineering Group
Computational Science and Engineering Department
Rutherford Appleton Laboratory
shawn.chin@stfc.ac.uk

David Worth
Software Engineering Group
Computational Science and Engineering Department
Rutherford Appleton Laboratory
d.j.worth@rl.ac.uk

Chris Greenough
Software Engineering Group
Computational Science and Engineering Department
Rutherford Appleton Laboratory
c.greenough@rl.ac.uk

ABSTRACT

High performance computing is essential for simulating complex problems using agent-based modelling (ABM). Researchers are hindered by complexities of porting models on parallel platforms and time taken to run large simulations on a single machine. This paper presents FLAME framework, the only supercomputing framework which automatically produces parallelisable code to execute on different parallel hardware architectures. FLAME's inherent parallelism allows large number of agents to be simulated in less time than comparable simulation frameworks. The framework also handles the parallelisation of model code allowing modellers to run simulations on number of supported architectures.

The framework has been well tested in various disciplines like biology and economics projects yielding successful research results like project EURACE, where the European economy was modelled using agents. More recently FLAME has been ported to consumer NVIDIA Graphics Processing Units (GPUs) allowing parallel performance equal to that of grid architectures with the ability to perform real time visualisation.

Categories and Subject Descriptors

I.3.1 [**Computer Graphics**]: Hardware Architecture-Parallel processing, Graphics processor

General Terms

Performance

Keywords

Simulation techniques, tools and environments

1. INTRODUCTION

FLAME (Flexible Large-scale Agent-based Modelling Environment) is a agent-based modelling framework which allows modellers from various disciplines like economics, biology and social sciences to easily write agent-based models and simulate them on parallel hardware architectures [1]. The environment is a first of its kind which enables creation of agent-based models that can be run on high performance computers (HPCs) and GPUs. Being based on the logical communicating extended finite state machine theory (X-machine), gives agents more power enabling large complex systems to be simulated. The simulation code is generated by processing a model definition using a template engine. Given a set of predefined code templates and flags indicating the architecture, the template engine generates custom simulation code (including common routine for input/output) which can be compiled and executed. All algorithms and parallel code are contained within the templates themselves abstracting this from the modeller. Initial configurations of agents are expressed in XML format and are used as a program argument to set up the initial agent population. Figure 1 depicts these as a block diagram for the FLAME architecture. The libmboard file is a specific file written to access messages using message passing interface (MPI) on parallel computers. The template engine takes input the XML model description and agent functions. The initial agent population settings are set in 0.xml file read during the simulations.

Communication between agents is achieved through the use of messages. In case of HPC, FLAME uses MPI to port messages across different machines and platforms. The communication is handled using an intelligent message board library, Libmboard, which allows filtering of messages reducing the work for the agents improving simulation performances. Using a distributed memory model, single program multiple data (SPMD), the framework is able to han-

---

[*] Contact for Flame www.flame.ac.uk

[†] Contact for Flame GPU www.flamegpu.com

Cite as: FLAME: Simulating Large Populations of Agents on Parallel Hardware Architectures, Mariam Kiran, Paul Richmond, Mike Holcombe, Lee Shawn Chin, David Worth and Chris Greenough, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1633-1636

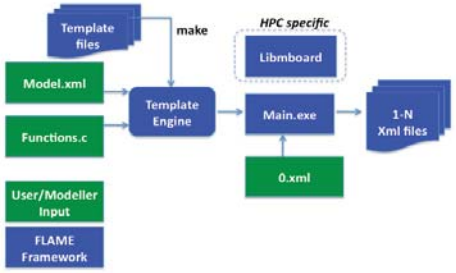

Figure 1: Block diagram for FLAME simulation program.

dle deadlocks through synchronisation points which ensure all data is coordinated among agents. In the case of the GPU simulation, distributed message communication is not required as all data is maintained on a single device. Instead a number of highly optimised communication algorithms are defined allowing optimal performance to be achieved.

2. FLAME AND HPC: SIMULATING EURACE

The EURACE project involved simulating the European economy with various markets interleaved together, namely the labour, investment goods, credit and financial markets. The project involved running simulations to the order of a million (10^6) agents to simulate an accurate depiction of the European economy.

One of the biggest challenges faced during the project was simulating and porting large numbers of agents on different parallel computing platforms. Some of the machines available were some of the fastest parallel supercomputers in the world namely HPCx, SCARF and HAPU. This presented challenges for anticipating and handling of the communication dependencies among the agents. To achieve efficient communication among the agents, MPI libraries were used as a message board library. This provided various options which help speed up the processing time in parallel computers. These enhancements include filtering of messages for each agent on a criteria, sorting of message lists based on specific variables, randomisation of messages in certain situations and using iterators to add intelligence to the manner in which message boards are handled.

Using agent-based modelling for research in stochastic systems allows to represent great detail removing most assumptions used in differential equations. However being used for research by biologists or economists, most modelling frameworks force the users to learn computer languages to write their models. Additionally if advanced work like parallel simulations are performed, it makes it very difficult for noncomputer professionals to use ABM research. FLAME overcomes this issue by using a basic easy to understand XML schema to write model description and a simple C language to write agent functions. It is the responsibility of the FLAME framework to run the code in serial or parallel as the modeller intends to by using simple command line options which have been well documented in FLAME user manuals and example files. A complex EURACE FLAME model was ran on various machines giving promising results for economics as well as computing advancements, drawing suitable conclusions on various economic policies applied in the EU and other aspects like the inclusion of new EU members affecting the European economy [2].

3. FLAME AND GPU

In addition to HPC, a version of FLAME has been ported to the Graphics Processing Unit (GPU) [4]. Although this version differs in the type of parallelism (data parallelism is used rather than task parallelism) the syntax for model and behaviour scripting remains the same allowing models to be compiled for either architecture.

Technically the GPU not only exceeds the transistor count of modern CPUs, but a significantly higher portion of transistors are available for data processing rather than data caching and flow control. In addition to this the GPUs memory bandwidth exceeds that of system memory bandwidth by roughly a factor of 10. Recently GPU computing has benefited from the introduction of improved programming interfaces implemented by hardware vendors making the architecture more accessible. Despite this, performance gains are often achieved only through careful optimisation, requiring advanced knowledge of the hardware's capabilities and optimal operating conditions.

Using the template driven FLAME simulation technique, templates for generating GPU simulation code makes ABM on the GPU accessible to modellers without an understanding of complex GPU programming. The advantages of this include a method of high performance ABM which is available using consumer hardware. Performance rates equalling or bettering that of HPC clusters can easily be achieved with obvious cost to performance benefits. Massive population sizes can be simulated far exceeding those which can be computed (in reasonable time constraints) within traditional ABM toolkits. The use of data parallel methods ensures that the techniques used are applicable to emerging multi-core and data parallel architectures. Finally the use of the GPU allows ABMs to be visualised in real time. This is a result of both the speedup achieved and the avoidance of transfer bandwidth costs by maintaining model data on the GPU device. Within our demonstration a number of specific ABM case studies are shown which include swarm based systems and cellular level biological modelling (with highly accurate inter agent force resolution) [3]. In all cases real time (or faster) simulation performance is achieved and combined with visualisation to allow run-time manipulation of simulation parameters. This real time manipulation is particularly useful for exploration of modelling parameters and simple visual (face) validation.

4. CONCLUSION AND FUTURE WORK

The FLAME framework offers significant performance advantages for ABM allowing simulations at scales which were previously inconceivable. The ability to exploit very different parallel architectures allows a wide range and scale of models to be simulated. The framework, in HPC and the GPU version, abstracts the underlying hardware offering flexibility as a unique attempt to high performance ABM. Future improvements with FLAME for HPC include dynamic agent load balancing and writing out huge simulation files containing the results of the simulations. Various new technologies such as HDF5 formats are being used to enhance these facets of FLAME. It is also hoped that the use of the GPU will be explored for models such as EURACE which are currently to large (in their memory requirements) for GPU implementation. This will most likely use multiple GPU devices and the HPC message passing libraries to provide communication between these.

(Demo URL can be found at: http://www.youtube.com/profile?user=prichmonduk#p/u/0/gaO0JRM9x5s)

5. REFERENCES

[1] S. T. Coakley, R. Smallwood, and M. Holcombe. From molecules to insect communities - How formal agent based computational modelling is uncovering new biological facts. *Scientiae Mathematicae Japonicae Online*, pages 765–778, March 2006.

[2] C. Deissenberg, S. van der Hoog, , and H. Dawid. Eurace: A massively parallel agent-based model of the european economy. *Applied Mathematics and Computation*, 204:541–552, 2008.

[3] P. Richmond, S. Coakley, and D. Romano. Cellular level agent based modelling on the graphics processing unit. In *In Prceedings of the Workshop on High Performance Systems Biology 2009 (HiBi09). To appear*, 2009.

[4] P. Richmond, S. Coakley, and D. Romano. A high performance agent based modelling framework on graphics card hardware with cuda. *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS), 2009, Budapest, Hungary*, 2009.