



Project no.
035086

Project acronym
EURACE

Project title
An Agent-Based software platform for European economic policy design with heterogeneous interacting agents: new insights from a bottom up approach to economic modelling and simulation

Instrument: STREP

Thematic Priority: IST FET PROACTIVE INITIATIVE "SIMULATING EMERGENT PROPERTIES IN COMPLEX SYSTEMS"

Deliverable reference number and title
D1.4: Porting of agent models to parallel computers

Due date of deliverable:
31/08/2008

Actual submission date:

Start date of project: September 1st 2006

Duration: 36 months

Organisation name of lead contractor for this deliverable
STFC Rutherford Appleton Laboratory - STFC

Revision 1

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Contents

1	Introduction	4
2	General Parallel Implementation of FLAME	4
3	Conclusion	6

Abstract

Making use of high performance computers in agent-based simulation is a complex and difficult task. This report describes the approach being used within the EURACE project to exploit parallel computing technology in large-scale agent-based simulations involving millions of agents. The underlying software is the FLAME framework and the paper will give an overview of features and use of FLAME and discuss the implementation of techniques that attempt to exploit large parallel computing systems. Some early simulation results will be presented together with a discussion of the requirements for efficient parallel simulations and the performance of the current implementation

1 Introduction

In this report we described the parallel implementation of the FLAME Framework [5]. We will cover not only the reason for parallelisation but also a detailed description of the approach being adopted coupled with some initial results from a set of benchmarks which hope to capture the computation and communications loads inherent in the EURACE application.

Before considering the current parallelisation of FLAME it is worth considering the characteristics of a software system than make it worth taking the time and effort to parallelise. Some characteristics of when to parallelise:

- Code is practically incapable of running on one computer, memory requirements too great, run time too long
- Code will be reused frequently - parallelisation is a large investment
- Data structures are simple, calculations are local, easy to communicate and synchronize between processors

The converse should also be considered. When not to parallelise a code:

- Code will only be used once (or infrequently) - An efficient parallel code takes time to develop!
- Current performance is acceptable and execution time is short
- There will be frequent and significant code changes

It should also be remembered that some algorithms simply do not parallelise.

2 General Parallel Implementation of FLAME

Because of its architecture FLAME does have some of the good characteristics but unfortunately it has a number of the bad. In the context of the EURACE project we believe that the good out-weigh the bad with the size and time to solution dominating the good. However in a fully connected and communicating agent population the communications may not be local but long range. However in many applications of FLAME, EURACE included, we have seen that there is sufficient locality to consider parallelisation given the general population sizes.

The use of simple read/write, single-type message boards allows the framework implementer to divide the agent population and their associated communications areas. This division could be based on any number of parameters or separators but the simplest to appreciate is position or locality. If, as in EURACE, agents are people or companies for example, they will have locality defined either as location or by some group topology. It is also reasonable to assume that the dominant communications in both scenarios will be with neighbouring agents.

As explained above FLAME uses a collection of message boards to facilitate inter-agent communication. As the majority of large high performance computing systems currently use a distributed memory model a Single Program Multiple Data (SPMD) paradigm is considered most appropriate for the FLAME architecture. The parallelisation of FLAME utilises partitioned agent populations and distributed message boards linked through MPI communication. Figure 1 shows the difference between the serial and parallel implementation.

The most significant operation in the parallel implementation is providing the message information required by agents on one node of the processor array but stored on a remote node of the processor. The FLAME Message Board Library manages these data requests by using a set of predefined message filters to limit the message movement. This process could be considered

a synchronisation of the local message boards within an iteration of the simulation. This synchronisation essentially ensures that local agents have the message information they need as the simulation progresses. The two main areas of algorithmic and technical development needed to

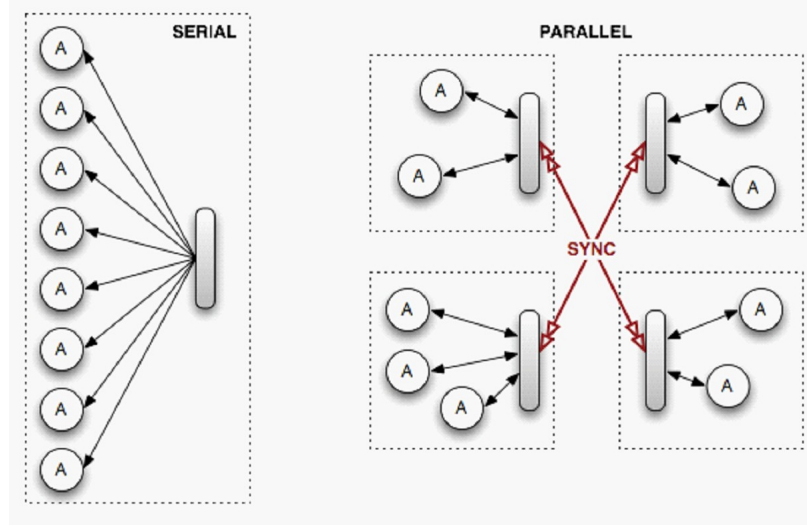


Figure 1: Serial and Parallel Message Boards

achieve an effective parallel implementation are: load balancing and communications strategy.

Initial load balancing is not too difficult: we have a population of agents - of various complexities to which we can assign relative weights - in the most general case the agents can be distributed over the available processors using the weights. This may well give an initial load balance but make no reference to the possible communications patterns of the agent population. As the simulation develops the numbers of agents in the population may change and adversely affect the load balance of the processors. It is a very interesting and difficult problem to gauge whether the additional work (computation and communication) involved in remedying a load imbalance is worth the gain. Given that the goal of any dynamic re-organising of the agents is to reduce the elapsed time of the overall simulation, determining whether a process of dynamically re-balancing the population will contribute to this is very problematic. It may well be that a slight load imbalance will have no significant effect on the wall clock time of the simulation. These problems are under investigation.

The communications patterns and volumes of the population will have a considerable impact on the performance and parallel efficiency of the simulation. In general agents are rather lightweight in the computational load. Where all agents can and do communicate with all others the communications load within and across processors will be great. Fortunately communications within a processor are generally efficient. However across processors this communication can dominate the application. Within FLAME communications between agents is managed by the Message Board Library, which uses MPI to communicate between processors. The Message Board Library implementation attempts to minimise this communication overhead by overlapping the computational load of the agents with the communication.

Where the agents have some form of locality the initial distribution of agents makes use of this information in placing agents on processing nodes. During the simulation agents can be dynamically re-distributed to maintain computational load balance. However given the lightweight computational nature of many agent types the effect of dynamically re-distributing agents on the grounds of their communications load may well turn out to be more important than computational load.

Within the EURACE Project a parallel version of FLAME has been developed using these

ideas and below are discussed some of the initial results in performing parallel simulations.

3 Conclusion

In this report we have described the parallel implementation of the FLAME framework and demonstrated its use in a number of EURACE related simulations including one of the complete EURACE model.

References

- [1] Tesfatsion (2006) "Agent-based computational economics: a constructive approach to economic theory" in Handbook for Computational Economics, Vol 2, Noth-Holland
- [2] Finin et al (1994) "KQML as an Agent Communication Language", The Proceedings of the Third International Conference on Information and Knowledge Management
- [3] Gregory et al (2001) "Computing Microbial Interactions and Communications in Real Life", 4th International Conference on Information Processing in Cells and Tissues
- [4] Noble (2002) "Modeling the heart-from genes to cells to the whole organ", Science
- [5] Coakley (2005) "Formal Software Architecture for Agent-Based Modelling in Biology", PhD Thesis, University of Sheffield
- [6] Walker et al (2004) "Agent-based computational modeling of wounded epithelial cell monolayers", IEEE Transactions in NanoBioscience
- [7] Walker et al (2004) "The Epitheliome: Agent-Based Modelling Of The Social Behaviour Of Cells", Biosystems
- [8] Pogson et al (2006) "Formal Agent-Based Modelling of Intracellular Chemical Reactions", to appear in Biosystems
- [9] Qwarnstrom et al (2006) "Predictive agent-based NFkB modelling - involvement of the actin cytoskeleton in pathway control", Submitted
- [10] Jackson et al (2004) "Trial geometry gives polarity to ant foraging networks", Nature
- [11] EURACE (2006) "Agent-based software platform for European economic policy design with heterogeneous interacting agents", EU IST Sixth Framework Programme.
- [12] Holcombe (1998) "X-machines a basis for dynamic system specification", Software Engineering Journal
- [13] Kefalas et al (2003) "Communicating X-machines: From Theory to Practice", Lecture Notes in Computer Science
- [14] Kefalas et al (2003) "Simulation and verification of P systems through communicating X-machines", Biosystems
- [15] Eleftherakis et al (2003) "An agile formal development methodology", Proceedings of the First South-East European Workshop on Formal Methods
- [16] Delli Gatti et al (2006), "Emergent Macroeconomics An Agent-based Approach to Business Fluctuations", submitted.