Environmental Modelling & Software 24 (2009) 819–831

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

A software framework for fine grain parallelization of cellular models with OpenMP: Application to fire spread

Eric Innocenti [a], Xavier Silvani [a], Alexandre Muzy [a], David R.C. Hill [b,*]

[a] University of Corsica, SPE – UMR CNRS 6134, B.P. 52, Campus Grossetti 20250 Corti, France
[b] Blaise Pascal University, ISIMA/LIMOS UMR CNRS 6158, BP 10125, Campus des Cézeaux 63177Aubière Cedex, France

ARTICLE INFO

Article history:
Received 10 November 2006
Received in revised form
13 November 2008
Accepted 18 November 2008
Available online 5 February 2009

Keywords:
Software framework
Fire spread physical model
DEVS
Open MultiProcessing (OpenMP)
Symmetric multiprocessors (SMP)

ABSTRACT

We are dealing here with the parallelization of fire spreading simulations following detailed physical experiments. The proposal presented in this paper has been tested and evaluated in collaboration with physicists to meet their requirements in terms of both performance and precision. For this purpose, an object-oriented framework using two abstraction levels has been developed. A first level considers the simulation as a global phenomenon which evolves in space and time. A local level describes the phenomena occurring on elementary parts of the domain. In order to develop an extensible and modular architecture, the cellular automata paradigm, the DEVS discrete event system formalism and design patterns have been used. Simulation treatments are limited to a set of active elements to improve execution times. A new kind of model, called Active-DEVS is then specified. The model is computed with a fine grain parallelization very efficient for present day multi-core processors which are elementary units of modern computing clusters and computing grids. In this paper, the parallelization with Open MultiProcessing (OpenMP) standard directives on Symmetric MultiProcessing (SMP) architectures is discussed and the efficiency of the retained solution is studied.

Software availability

Name of software: The C++ code is freely available with a simple request to Eric Innocenti ino@univ-corse.fr.

1. Introduction

Simulation is a powerful tool that enables a better understanding of real world problems. Among the various simulation techniques, the simulation of discretized differential equations has been used to describe complex systems and interpret many real world problems. To deal with space, these simulation models are usually represented as cell spaces (Wu et al., 2004; Langlois and Phipps, 1997; Karafyllidis and Thanailakis, 1997).

Fire spreading is a good example of complex phenomena that relies on differential equations and cellular models. In fire spread modeling, discrete equations are written down from the conservative laws for mass, momentum and energy that govern the system. In this paper, the domain context is multi-scale, from small-scale laboratory experiments to field scale experiments, hence, the physical complexity of the thermodynamic system under consideration strongly increases. In actual fires, a set of new control parameters and distributions appear in comparison with laboratory fires. These are mainly governed by a multi-scale structured vegetation (shrub or forest), the turbulent flow regime of the flame front and its crossing wind flow and the fractal nature of the topography. This enlargement in the range of scales (e.g. from few centimeters of a leave to several kilometers) supposes huge computing resources in order to capture each representative scale governing the conservative laws. The numerical resolution of the partial differential equations forming the system of conservative laws for a reactive fluid flow is known as Computational Fluid Dynamics (CFD), including multiphase flow simulations (Morvan and Dupuy, 2004) and Large Eddy simulations (Mell et al., 2007).

An efficient alternative to CFD for fire simulation is the cellular automata approach. Indeed, cellular automata appear to be appropriate for modeling such complex spatial phenomena as large-scale fires, because of their discrete nature and their suitability for implementation on computers (Tian and Burrage, 2005; Lay, 2000) The methods ensuing from this paradigm emphasize local interactions as opposed to a global description. The generated emergent behaviors are often surprisingly complex. Wolfram and others have shown such emergent features of cellular automata (Wolfram, 2002; Talia, 2000). In fire spread modeling, this was

* Corresponding author. Tel.: +33 0 473 40 50 19.
E-mail address: drch@isima.fr (D.R.C. Hill).

1364-8152/$ – see front matter © 2008 Published by Elsevier Ltd.
doi:10.1016/j.envsoft.2008.11.014

recently illustrated in (Porterie et al., 2007) comparing a cellular automata approach to experiments performed at the laboratory scale. At a field scale, in a strategy involving cellular automata, the coupling with atmospheric models and topographical models (stored in a Geographical Information System) shall be planned (Clark et al., 2004.) Furthermore, some new trends in fire safety research include modern data assimilation techniques (Cohen et al., 2007). All these new modular components of a simulator for wildfire at a field scale necessitate using techniques for high performance computing (the coarse-grained parallelization for data assimilation or atmospheric models). At the end, prediction tools of a wildfire behavior at fields scale require simulation times shorter than real times for an optimal management of firefighting.

Therefore, fire spreading modeling and simulation require efficient models and very high performance computations. The efficiency of discrete event simulators is beginning to receive significant attention (Wainer and Giambiasi, 2001; Lee and Kim, 2003; Hu and Zeigler, 2004; Muzy and Nutaro, 2005). Concerning cellular models, (Wainer and Giambiasi, 2001) show that the simulation of cellular models can be improved by "flattening" the hierarchy of coordinator objects which are used in the DEVS (Discrete EVent System Specification) (Zeigler, 1976; Zeigler et al., 2000a,b). In this flattened simulator, a single coordinator manages all the atomic components of a model. This can significantly reduce the cost of event routing, and it eliminates the need for multiple coordinator objects. Lee and Kim (2003) describe a similar solution that computes and stores possible event routes at compile time. Hu and Zeigler (2004) describe an improved scheduling algorithm for cellular models that are simulated using a hierarchy of coordinators and simulators. Muzy and Nutaro (2005) proposed a new simulation approach for DEVS models (Zeigler et al., 2000a,b) and Parallel Dynamic Structure Discrete Event (DSDEVS) models (Barros, 1997). The simulation architecture and communication protocol have been designed to improve efficiency by: (i) eliminating unnecessary simulator and coordinator objects, (ii) faster event scheduling by only storing references to active models, (iii) eliminating unnecessary internal synchronization messages, and (iv) eliminating unnecessary event routing messages.

From a modeling point of view, three kinds of approach have been developed so far through the DEVS formalism:

1. "Pure" DEVS models, in which cells are specified and simulated as usual atomic models (see Ntaimo et al., 2004) for a good example on fire spread),
2. A higher specification level has been developed through Cell-DEVS (Wainer, 2002). Timing mechanisms abstractions and geometrical sets have been added to usual DEVS structures,
3. Non-modular approaches to improve simulation efficiency (Muzy et al., 2003; Shiginah, 2006). In the latter very sound proofs of closure under coupling are provided. A simplified specification level is provided to facilitate modeling and improve simulation performances.

The third approach is definitely chosen from a modeling point of view (using a non-modular approach) and a simulation point of view, using simplified aggregated simulators: (Muzy et al., 2003). Furthermore, our fire spread model has already been simulated and modeled through the Cell-DEVS formalism (Muzy et al., 2002a,b, 2005; Wainer, 2006). The implementation was performed but we were not able to provide reasonable simulation times compared to the physical propagation time. Indeed, this fire spread model necessitates very small time and space discretizations leading to important computation overheads. Two reasons can explain this:

1. As B.P. Zeigler (DEVS' father) says: "(…) rule-based model specification (…) is achieved at some cost in execution time" (Zeigler, 1990). Moreover, dealing with discrete event timing mechanisms at a low level (here the cells), and for discrete-time simulation, leads to data structure overheads due to message exchanges. The next point justifies this overhead with comments from Shiginah, one of Bernie Zeigler's students:
2. "Cell-DEVS formalism, since it represents each cell as an atomic model, is considered as a conventional DEVS implementation of cell space models which has the performance drawback that is resulted by the huge volume of inter-cell communication generated during simulation. In addition, expressing cellular models in Cell-DEVS formalism is, to some extent, complex and requires more efforts at the modeler level. On the other hand, this dissertation introduces the multi-layer approach to simplify the modeling process and make the cell space's extensive specifications transparent to the end user." (Shiginah, 2006).

Hence, Active-DEVS aims to be an efficient model for simulating large-scale propagation phenomena. The software tools implemented have to keep pace with the rapid improvements of processing power of SMP machines. To reach these objectives, a cellular model (Worsch, 1997) and a simulation framework, founded on both DEVS (Zeigler et al., 2000a,b) formalism and object-oriented methodology, have been used.

An enhanced automaton models the propagation domain in which elementary behaviors describe each node. The spatial dynamics expression of the phenomena is thus facilitated. The DEVS formalism simplifies the modeling at a higher level and allows specifying components independently from automatically generated simulation algorithms. Hence, descriptions of the simulation treatments are facilitated.

The object-oriented architecture relies on design patterns and thus keeps a modular, elegant and adaptable design (Gamma et al., 1994.) A specific approach founded on an object container integrating parallel directives for SMP machines is performed. The implementation of this local parallelization allows improving execution times and is reported as fine-grained parallelization. The software simulation tool developed combines the experienced features of cellular automata, object-oriented methodology (design patterns), DEVS, and the efficiency of parallelization techniques based on OpenMP parallel compiler directives (OpenMP, 2002.) The presented approach is designed to evolve for predicting wildfires at field scale. This study therefore attempts evaluating the efficiency of the chosen fine-grained parallelization when combined with a modern object-oriented architecture, with cellular automata.

On the one hand, modeling is facilitated by the use of object-oriented techniques and DEVS. This technique and this framework have been used successfully in many different domains (for a review on DEVS applications, please refer to Zeigler et al., 2000a,b.) At the simulation level, design patterns are used to switch easily between different data structure implementations depending on cellular modeling requirements (more details are provided in sub-section 3.1) and parallelization techniques.

The main contributions of this approach can be summed-up as follows:

- A physics-based model is implemented in a consistent and adapted object-oriented and formal framework,
- The latest advances in the domain of discrete event modeling and simulation are used for optimization (at both modeling and implementation levels),
- SMP parallel implementations are included in a generic way,