Parallel Discrete Event Simulation Performance Modeling and Evaluation

Overeinder, B.J.; Sloot, P.M.A.

Publication date
1995

Parallel Discrete Event Simulation Performance Modeling and Evaluation

Benno Overeinder and Peter Sloot
Parallel Scientific Computation & Simulation Group
University of Amsterdam

Keywords parallel discrete event simulation, performance modeling, performance evaluation

Introduction

Large discrete event simulations are known to consume significant amounts of time on sequential machines. One basic approach to the problem of reducing the required simulation time is the exploitation of parallelism. However, in parallelizing the simulation new problems arise. Due to the distributed generation of events causality errors can occur. As a result the sequence in which to process the events is essentially indeterminate. Several methods have been developed to avoid or to circumvent these causality errors. One of the methods which seems to offer the greatest potential as a general purpose simulation mechanism is the Time Warp paradigm [1].

To predict the performance metrics of parallel execution of discrete event simulations, tools have been developed to analyze the sequential execution of these simulations and to give bounds on the performance. To that end, an average parallelism evaluation tool is developed and implemented to obtain a conceptual simple, but very meaningful characterization of the software system and the bounds on the performance. These performance metrics can also be applied as a measure in the evaluation of effectiveness of the Time Warp paradigm in parallel simulation. An implementation of the Time Warp paradigm is realized on the Parsytec GCel-3/512 as well as on the Parsytec PowerXplorer under the PARIX operating system.

The Time Warp Performance Evaluation Environment

We are specifically interested in the application of the Time Warp method to dynamic complex systems that can be modelled with Asynchronous Cellular Automata (ACA) [2]. In addition to the modelling and parallel simulation of ACA, we are also interested in the performance prediction and evaluation of the simulation system. To that end, we formulated a benchmark environment that provides a platform for experimentation with ACA. The benchmark environment, aside of the ACA computational model and the Time Warp protocol, is composed of the following parts: a performance model, a parallelism analysis tool, and performance measurement tool (see Fig. 1).

A performance model of the ACA and the Time Warp method is constructed for the prediction of the parallel performance of the ACA with the Time Warp method. We describe the stochastic characteristics of the components of the system with use of Markov modelling and measure their response to well defined stimuli.

The use of the critical path analysis tool is twofold. First, it can be used within the validation and verification of the standalone ACA performance model. The ACA performance model gives a prediction about the parallel performance, and the critical path analysis tool actually determines the potential parallelism, i.e., the parallelism inherent to the application. Second, this measured

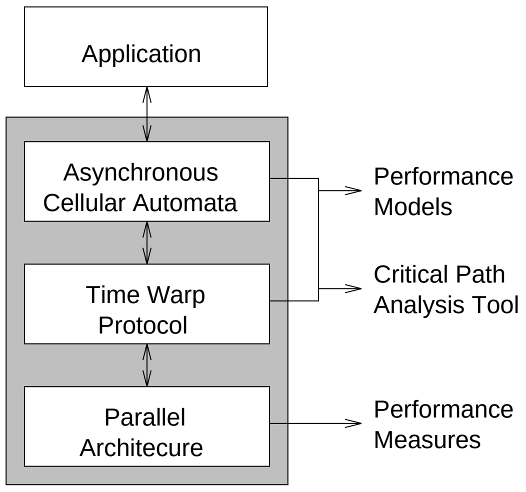


Figure 1: The simulation environment.

inherent parallelism is used in the evaluation of the efficiency and effectiveness of the parallel simulation method. The performance metrics measured after execution of the ACA with Time Warp can be compared with the results of the critical path analysis tool. In this way it is possible to see how much of the potential parallelism in the application is actually realized. This allows for fine tuning of the simulation environment to the parallel platform.

The performance model of the ACA and the Time Warp method can than be verified by performance measurements extracted from the execution of the simulation system. This will result in a better understanding whether an application can be effectively solved by an ACA in combination with the Time Warp mechanism.

Critical Path Analysis and Ising Spin System Simulation

Critical Path Analysis

In performance analysis and evaluation of Parallel Discrete Event Simulation (PDES) protocols we need a measure to compare the effectiveness of the different protocols. A *relative* criterion can be obtain by comparing the measured execution time of the different protocols. This allows for a ranking of the different protocols, but does not answer the question: “How much of the inherent parallelism is actually realized and what is the lower bound on the execution time?” This lower bound on the execution time is the ultimate goal that the PDES protocols strive to, and not their relative position among the others. An *absolute* measure that expresses the ability of the PDES protocol to exploit the available parallelism would be very useful in the evaluation of the various protocols.

A discrete event simulation can be described by a program dependency graph, also called a space-time diagram (see Fig. 2). In the program dependency graph, the concurrent entities are identified and their intra- and interdependencies described by directed arcs in the graph. The lower bound on execution time (and the inherent parallelism) can be computed from the critical path in the space-time diagram [3].

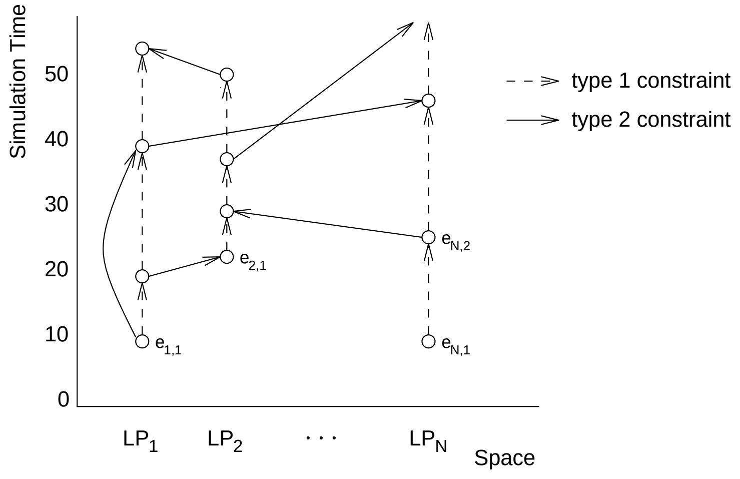


Figure 2: Space-time diagram depicting the events with their dependency constraints.

Ising Spin System Simulation

For our example, we will look into the two-dimensional version of the Ising spin system. The model is comprised of an $n \times n$ square lattice in which each lattice site has an attribute called *spin*, which can take on either of the values 1 (known as up spin) or -1 (known as down spin). Spins on adjacent, nearest-neighbor sites interact in a pair-wise manner with a strength J (known as the exchange energy). There may also be an external field of strength B (known as the magnetic field). The magnetization of the system is the difference between the number of up and down spins on the lattice.

The parallel implementation of the Ising spin system exploits data parallelism available in the system. The lattice is subdivided in sub-lattices of equal sizes, and distributed over the parallel processors.

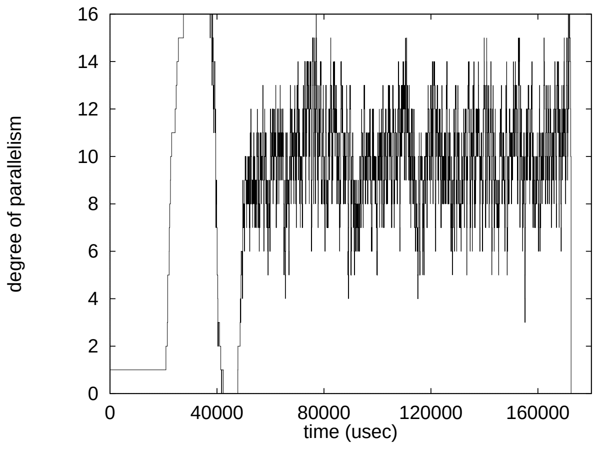
Sample Run with Critical Path Analysis Tool

From the execution of the Ising spin system with the Time Warp protocol on sixteen nodes of the Parsytec PowerXplorer, we have extracted the following trace data.

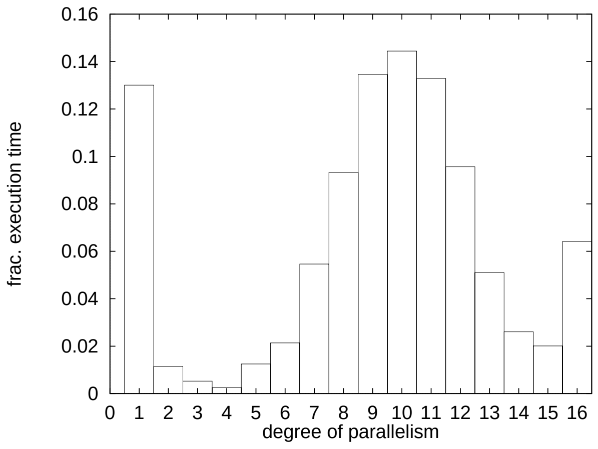
In Fig. 3(a) the profile of the execution of the simulation is shown. During the execution, the number of active processors vary between one and sixteen. One can clearly identify an initialization phase in the beginning of the simulation. This corresponds to the exchange of boundary information, required by the Ising spin implementation. After the initialization phase, the simulation continues with randomly updating lattice points, and exchanging message when boundary points are flipped.

Figure 3(b) is a histogram of the degree of parallelism during the execution of the Ising spin simulation. From the shape of the execution, one can see that 13 % of the time only one process is active (most prominent in the initialization phase of the simulation). The other values are clustered around the degree of parallelism of ten. This corresponds with the second phase of the execution in Fig. 3(a). The average parallelism, A, for this simulation run is 8.82, while the measured speedup, S, is 1.90.

The abstraction of the parallel execution to the space-time diagram and the critical path analysis provides information for performance "debugging" and directs users to bottlenecks in the program. This can be very useful in comparing different decompositions of the same problem by showing how



(a) Profile of the execution.



(b) Shape of the execution.

Figure 3: Trace output of Ising spin simulation.

much concurrency is available in each decomposition.

References

[1] D. R. Jefferson, "Virtual Time," *ACM Transactions on Programming Languages and Systems*, vol. 7, no. 3, pp. 404–425, July 1985.

[2] B. J. Overeinder and P. M. A. Sloot, "Application of Time Warp to parallel simulations with asynchronous cellular automata," in *Proceedings of the 1993 European Simulation Symposium*, (Delft, The Netherlands), pp. 397–402, Oct. 1993.

[3] B. J. Overeinder and P. M. A. Sloot, "Parallel performance evaluation through critical path analysis," in *Proceedings of the HPCN Europe 95 Conference*, (Milan, Italy), Springer Verlag, May 1995.

Contact address

Contact:	B. J. Overeinder
Institution:	Parallel Scientific Computation & Simulation Group
	Department of Computer Science, University of Amsterdam
Mail:	B. J. Overeinder
	Parallel Scientific Computation & Simulation Group
	Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
Phone:	+31 20 525 7463
Fax:	+31 20 525 7490
Email:	bjo@fwi.uva.nl
WWW:	http://www.fwi.uva.nl/fwi/research/vg4/pwrs/