Parallel and Distributed Simulation

Alois Ferscha
Institut für Angewandte Informatik und Informationssysteme
Universität Wien
Lenaugasse 2/8, A-1080 Vienna, Austria
ferscha@ani.univie.ac.at

Parallel and distributed simulation (PDS) over the one and a half decade of its existence has turned out to be more foundational than merely solving the causality preservation dilemma in partially ordered event structures as they occur in parallel and distributed simulation executions. Today’s availability of parallel and distributed computing and communication technology has given a new relevance to the field that could not have been foreseen in its early days. The potential improvement of elapsed time for large simulation experiments via the involvement of a set of individual processing nodes of a parallel (shared or distributed memory) computer or distributed system is more promising today than at any time in history of the field. Accelerating simulation experiments for large system models is naturally the outstanding PDS research goal. But above this, many scientists, also from within the core of classical PDS field, have seen impact of the PDS theory and methodology also in nonstandard application areas such as parallel program execution.

The intent of this minitrack was to provide a forum for the exploration of new high performance simulation concepts and techniques, as well as their successful application on today’s and tomorrow’s parallel execution platforms.

Twenty-six contributions have been submitted by authors from France, Germany, Japan, Poland, Sweden, UK and the USA, after more than seventy authors announced papers via abstracts. Eighty-four reviewers with their expertise helped to select ten full papers and two short papers for publication in the proceedings, and for oral presentation at the conference. Each paper was reviewed by at least three, at most six, and on average by 3.96 reviewers. Due to the availability of papers in electronic format, the reviewing process could be managed almost exclusively by email.

A conceptual framework and an environment for high performance simulation is presented in the first paper. Zeigler, Moon, D. Kim and J.G. Kim propose the DEVS formalism for system modeling, and a DEVS implementation based on object oriented technology enabling a full exploitation of a parallel simulation model execution. DEVS modeling is illustrated in the context of wildland fire simulation, performance comparisons are conducted for a watershed model executing sequentially on a Sparc-1000 and in parallel on a CM-5.

The paper by Konas presents potentials of simulation at the confluence of object oriented system design and parallel processing: modularity, extensibility and reusability provide a natural and well structured approach to the construction of complex simulation models, while at the same time promoting a higher execution efficiency on a parallel platform.

Young and Wilsey propose a new distributed fossil collection technique for the Time Warp distributed discrete event simulation protocol. Basically, every fossil collector associated with a logical process (LP) by observing event arrival times establishes a statistical model for rollback distances to determine — in conjunction with a user defined risk factor that controls the aggressiveness of fossil collectors — a probabilistic GVT bound. The approach appears beneficial for the reduction of the amount of used memory over GVT based fossil collection, but requires additional checkpointing for possible “catastrophic” rollbacks, i.e. restoration of states that have been fossil collected due to an overestimation of the actual GVT.

Rönnegren, Barriga and Ayani have developed a benchmark suite for the performance evaluation of parallel simulation kernels on different architectures and the scalability analysis of certain simulation problems. It appears particularly hard to isolate performance influences stemming from the kernel as such or from the event structure underlying the simulation model executed by the kernel. Trying to abstract as much as possible from the latter performance impact, the authors construct a synthetic benchmark scalable
in the number of LPs, the state and message size, and the number of messages generated by an LP. As an illustration, the migration of a Time Warp kernel from a Sequent Symmetry S81 to a Sun Sparc SC2000 is reported.

**WARPED**, a Time Warp kernel implementation written in C++ and utilizing MPI, is offered as free software for experimentation and application development by Martin, McBrayer, and Wilsey. Ports of the software to various platforms are provided.

An analytical performance model respecting architecture specific communication and computation time was developed by Teo and Tay for the prediction of the elapsed simulation time and attainable speedup under various traffic load conditions of a conservative parallel simulation. Horizontal, vertical and modular partitioning schemes for simulation models of finite-buffer multistage interconnection networks are developed, and predicted simulation performance is compared to measurements. For a PVM-based implementation the authors find that performance tuning of message allocation and message buffering is more effective than network transmission speed improvement.

Simulation model partitioning and mapping issues, but in the context of VLSI circuit designs, are also addressed in the paper by Gerd Meister. A parallel VHDL simulator (DVSIM) has been developed with the goal to investigate performance characteristics of different parallel simulation algorithms, i.e. conservative with deadlock detection and recovery, as well as Time Warp. Four different gate-level LP packing and mapping strategies are systematically compared on circuits out of the ISCAS89 benchmark suite.

The paper of Weatherly et al. surveys ALSP (the Aggregate Level Simulation Protocol), i.e. the attempt to parallelize the DoD’s “advanced distributed simulation” initiative towards integrating simulation models, existing in a distributed environment, for cooperative simulations. ALSP, besides DIS, has become one of the largest (working) applications of distributed discrete event simulation methodology.

Kalantery in his paper tries to express a parallel simulators ability to complete a certain amount of activities in the simulation model within a certain period of real time as the “real time speed”. This ratio of a simulated time period to the real time period the simulator spent for event execution is considered of particular interest in telecommunication network simulations. Experiments with a range of SS7 models demonstrate stable and scalable real time performance of a conservative deadlock avoidance synchronization.

Dickens, Haines, Mehrotra and Nicol have elaborated a direct execution simulation engine for the execution of message passing parallel codes on a “virtual” parallel machine. For the scalability analysis and performance prediction of large parallel programs, where the number of involved virtual processors might exceed the number of physically available processors (by far), an environment is provided that superimposes application code execution by a simulation. A physical parallel machine executes a virtual parallel machine in that a set of virtual processors are mapped onto lightweight threads executing on a single physical processor. The authors manage to create separate address spaces for threads by “hiding” global data and functions in a global C++ class, an instance of which is created by each thread. Experimental results show that higher efficiency can be gained from a thread-based parallelization of the virtual machine simulator than from a (heavyweight, Unix-) process-based implementation.

Timed Petri Nets have proven to be a good modeling formalism for the quantitative (and qualitative) analysis of time dynamic, asynchronous, concurrent, discrete event systems. For the class of free choice nets that can be compositionally constructed from marked graph components via the insertion of “routing” places connecting those components, Gaual develops recurrence equations in the (min,+) algebra describing their evolution in time. Assuming the preselection (“routing”) policy for token reservation in the routing places, performance metrics like the total and average number of transition firings can be determined with fast, parallel algorithms. A time parallel, algebraic simulation is developed which by nature offers unbounded parallelism and appears very promising for very long simulation runs.

Finally, the paper by Weyer and Zheng extends their work on multi-level Newton methods for modified nodal circuit analysis involving nonlinear differential equations to “Quasi-Newton” techniques, that further reduce communication overhead in parallel circuit simulation on a workstation cluster.

The papers collected in this minitrack report on theoretical and methodological progress in the context of parallel and distributed simulation paradigms, and highlight new application areas of those concepts. Their results present a further step towards overcoming the perceived obstacles in making high performance simulation general purpose. I thank all the authors and referees for making this minitrack of high quality and topicality. My thanks also go to Jim Johnson and Gabi Kotsis, who helped out on critical days (and nights) to manage the administrative agenda.