Apr 14th, 12:00 AM - 12:00 AM

Applications of Parallel Discrete Event Simulation

Erik J. Jensen
Old Dominion University

APPLICATIONS OF PARALLEL DISCRETE EVENT SIMULATION

Erik J. Jensen
Department of Computational Modeling and Simulation Engineering
Old Dominion University
1300 Engineering and Computational Sciences Building, Norfolk, Va, USA
Ejens005@odu.edu

ABSTRACT

This work presents three applications of parallel discrete event simulation (PDES), which describe the motivation for and the benefits of using PDES, the kinds of synchronization algorithms that are used, and scaling behavior with these different synchronization algorithms.

Keywords: parallel discrete event simulation, time warp, simultaneous events, rollbacks.

1 INTRODUCTION

Parallel Discrete Event Simulation (PDES) is useful in applications for which logical processes with limited dependency can be distributed among many processing elements. PDES may be required due to memory constraints, and PDES might offer significant speedup compared with sequential discrete-event simulation due to increased cache hits. This paper discusses the findings of three works that use PDES.

2 WORKS

2.1 NeMo: A massively parallel discrete-event simulation model for neuromorphic architectures

Neuromorphic computing is a new type of processor technology that has some advantages over von Neumann processors [1]. Neuromorphic processors use far less power than von Neumann processors and excel at complex neural network computations with "a brain-like computational model" [1]. Energy efficiency is extremely important for HPC systems, as they strive for exascale. As a part of the Super-Neuro research project, the authors are developing "an architecture simulation framework [that combines] a number of modeling and simulation components to enable the design space exploration of potential hybrid CPU, GPU, and neuromorphic computer systems" [1]. *NeMo* is the neuromorphic architecture modeling component and can be "execute[d] in parallel using optimistic event scheduling… and reverse computation" [1]. *NeMo* is executed on "2,048 Blue Gene/Q nodes for a 8,388,608 core neuromorphic processor model" that generates about ten billion events per second [1]. *NeMo* is simulated on Rensselaer's Optimistic Simulation System (ROSS), which is an implementation of Jefferson's Time Warp algorithm that uses asynchronous event scheduling with rollbacks for out-of-order events [1]. Regarding simultaneous events, the Time Warp implementation on ROSS cannot guarantee deterministic event ordering in the case of simultaneous events, so in this study simultaneous events are avoided by "adding a random jitter factor [epsilon] to each event" [1]. Peak events per second (EPS) is achieved running 256 neurons per neurosynaptic core, meaning EPS decreases using 512 neurons per core as communication costs and rollbacks slowed performance [1]. Primary and secondary rollbacks increase precipitously as the number of Blue Gene nodes increase [1]. The 256-core real-time GVT simulation spends 45ms on primary rollbacks, and the 512-core real-time GVT simulation spends 2,308ms on primary rollbacks, about a 50x increase [1]. A figure shows many more secondary rollbacks, compared with primary rollbacks, and a faster-than-linear increase in the number of all rollbacks for NeMo2 benchmark tests as Blue Gene nodes increase [1]. The authors are interested in testing *NeMo* with a conservative event scheduler and also on a time-stepped simulation [1]. Another goal is to process simultaneous events without the jitter value [1].

2.2 Modeling and simulation of large-scale social networks using parallel discrete event simulation

Simulating online social networks (OSNs) can increase understanding of the spread of information and rumors, online influence, and privacy protection [2]. Parallel simulation is a good choice for modeling OSNs because "the underlying network structure often reaches terascale or petascale, which means the entire graph structure cannot fit into the main memory of a single computer" [2]. For fine-grained events, "the algorithms of social network analysis (SNA) often exhibit embarrassingly-parallel characteristics", but scalability is hindered by "inherent randomness and intensive communication of social dynamics" [2]. The authors have developed a framework, SUPE-Net that uses an "object-oriented parallel discrete event simulation (PDES) engine" for research of large-scale social networks on massively parallel architectures [2]. They evaluate the SUPE-Net framework with "the PageRank algorithm and the susceptible-infected-recovered (SIR) epidemic model on large-scale networks using part of the Tianhe-1A supercomputer" [2]. The nodes and links of the social network are distributed across multiple nodes and processors, and each node has an adjacency list that identifies neighboring nodes [2]. SUPE-Net is powered by the YH-SUPE "high performance, object-oriented" PDES engine [2]. Events are scheduled asynchronously, and either conservative or optimistic synchronization algorithms can be used [2]. The conservative algorithm uses lookahead, and the optimistic algorithm is Time Warp [2]. For the SIR model, the conservative algorithm, which uses zero lookahead, performs terribly in comparison to Time Warp, as cores increase [2]. The conservative algorithm in this case is not usable, but execution time generally decreases for Time Warp with scaling up to 150 cores [2]. There is no mention of simultaneous events or whether or not the simulations are deterministic.

2.3 PDES-A: Accelerators for Parallel Discrete Event Simulation Implemented on FPGAs

This work presents a design for an PDES accelerator called PDES-A implemented on a Field Programmable Gate Array (FPGA), which offers some benefits compared to PDES on traditional hardware [3]. One, "an FPGA can support fast and high-bandwidth on chip-communication, substantially alleviating the communication bottleneck that often limits the performance of PDES", and two, "a specialized accelerator… can more efficiently execute a required task without the unnecessary overheads of fetching instructions and moving data around a general datapath" [3]. In other applications, e.g. video encoding, this has demonstrated significant performance improvements and energy reductions [3]. PDES can potentially exploit the capabilities of FPGAs [3]. The FPGA simulator contains the event queue, event processors, system state memory, and the controller [3]. For a specific model, the object states and event-handling code are worked into a custom accelerator [3]. They run the PDES benchmark software PHOLD and find a "3.2x [improvement] with less than 15% of the power consumption" compared with ROSS on a Xeon machine [3].

REFERENCES

[1] M. Plagge, C. D. Carothers, E. Gonsiorowski and N. McGlohon, "Nemo: A massively parallel discrete-event simulation model for neuromorphic architectures," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 28, no. 4, pp. 1-25, 2018.

[2] B. Hou, Y. Yao, B. Wang and D. Liao, "Modeling and simulation of large-scale social networks using parallel discrete event simulation," *Simulation: Transactions of the Society for Modeling and Simulation International*, vol. 89, no. 10, pp. 1173-1183, 2013.

[3] S. Rahman, N. Abu-Ghazaleh and W. Najjar, "PDES-A: Accelerators for parallel discrete event simulation implemented on FPGAs," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 29, no. 2, pp. 1-25, 2019.