Introduction to the Special Issue on PADS 2019

This special issue highlights three articles extended from papers that appeared in the 2019 ACM Conference on Principles of Advanced Discrete Simulation (SIGSIM-PADS). It has been a tradition that *TOMACS* showcases some of the best papers from the SIGSIM flagship conference each year. This special issue is edited by Jason Liu and Laxmikant Kale, who served as the conference program chairs and were responsible for the proceedings of the conference. Kevin Jin was the general chair of the conference and helped with the initial organization of this special edition. He was excused from editorship due to conflict of interest. The articles appearing in this special issue have been selected also thanks to the help of Francesco Quaglia, who is the Editor-in-Chief of *ACM TOMACS*.

The outcome of the selection process identified six articles as possible candidates for this special issue, which are top ranked from reviews by the Technical Program Committee of SIGSIM-PADS 2019. The authors of three of them have been able to meet the deadlines for submitting the substantially extended versions of the original conference papers. To ensure quality, the extended manuscripts have been peer-reviewed by at least three reviewers, some through multiple rounds so the authors have the opportunity to carefully address the concerns and suggestions by the reviewers.

The special issue finally consists of three articles. The first article, entitled “[Distributed Virtual Time Based Synchronization for Simulation of Cyber-Physical Systems](#),” by Hannon, Yan, and Jin, presents a combined simulation and emulation testbed for cyber-physical systems (CPS) by integrating physical computing and networking components with simulation models using virtual time synchronization. More specifically, the authors propose two synchronization methods—periodic and dynamic—for an efficient implementation of a distributed virtual time on multiple embedded Linux devices. The article shows a use case of the proposed hybrid testbed for evaluating cyber-security of a voltage control application in the electric power grid. The original paper, from which this article is extended, won the 2019 SIGSIM-PADS Best Paper Award.

The second article, entitled “[Transitioning Spiking Neural Network Simulators to Heterogeneous Hardware](#),” by Nguyen, Andelfinger, Tan, Cai, and Knoll, presents a semi-automated method for transforming the CPU-based Spiking Neural Network (SNN) simulators so they can execute on heterogeneous hardware, including CPUs, GPUs, FPGAs, and DSPs. The proposed method allows one to write only a small number of core simulator functionalities common in SNN simulators, whereas a large portion of the model code is analyzed and transformed automatically. This method allows researchers to conduct new simulation studies on heterogeneous hardware in a timely manner, while still relying on widely studied and tested simulators and model implementations.

The third article, entitled “[Machine Learning Enabled Scalable Performance Prediction of Scientific Codes](#),” by Chennupati, Santhi, Romero, and Eidenbenz, presents a performance model for scientific applications on high-performance hardware architectures using machine learning and regression techniques. In particular, the authors present an analytical memory model that

ACM Reference format:

Jason Liu and Laxmikant Kale. 2021. Introduction to the Special Issue on PADS 2019. *ACM Trans. Model. Comput. Simul.* 31, 2, Article 9e (March 2021), 2 pages.
<https://doi.org/10.1145/3451235>

© 2021 Copyright held by the owner/author(s).

1049-3301/2021/03-ART9e

<https://doi.org/10.1145/3451235>

takes high-level application source code as input and predicts the execution time by analyzing the basic block structure of the code and architecture-independent virtual memory access patterns (such as memory reuse distance distribution) at the basic blocks and running detailed basic-block level simulations to determine hardware pipeline usage for overall runtime performance estimation of the application on the target architecture.

Jason Liu
Laxmikant Kale
Guest Editors