Tutorial: Lookahead, Rollback and Lookback, Quest for Parallelism in Discrete Event Simulation

Boleslaw Szymanski and Gilbert Chen
Rensselaer Polytechnic Institute, Troy, NY

Level: T2 (medium)
Goal: discussing state-of-the-art and presenting new developments in parallel discrete event simulation protocols
Orientation: simulation system developers and users

ABSTRACT:

Parallel Discrete Event Simulation (PDES) has long been regarded a hard problem, because the event granularity is usually so small that it is of little benefit to parallelize the execution of a single event. The only way to archive efficient parallelization is to execute events concurrently on different processors. Since the event dependencies are mostly dynamic, runtime identification of event dependencies to prevent causality errors becomes crucial to the success of the PDES technology.

Traditionally, two classes of synchronization protocols have been developed and extensively studied. Conservative protocols process events that are guaranteed to follow real-world causality order. As a result, these protocols are greatly limited by the availability of *lookahead*, the ability to predict the future. Optimistic protocols, on the other hand, achieve high degree of parallel processing by concurrent execution of events as soon as they become available. If causality errors occur, the system recovers from them by rolling back erroneously processed events. We will start with the description of these two classes of protocols and brief summary of their optimization techniques. We will then present a classical result that the speed of conservative simulations is limited by the length of the critical chain of events.

In our research, we observed that it is often easy to change the simulation past locally (on a single parallel processor), and named this ability lookback. The propagation of rollbacks can then be reduced by processing out-of-timestamp order events (stragglers) with a lookback procedure. In it, the event order is taken into account to correct the past. Furthermore, if each processor deliberately controls the pace of simulation to make sure that any stragglers received later can be safely processed by the lookback procedure, rollbacks can be completely eliminated. This leads to an entirely new conservative protocol that relies only upon the property of lookback. At the tutorial, we will define lookback and then discuss its relation to lookahead.

At first glance, lookback and lookahead seem to be two unrelated concepts, for they deal with distinct parts along the simulation time axis. It turned out that not only they are related, but also lookback incorporates

lookahead. When the simulation clock is more aggressively advanced regardless of potential causality errors, the future becomes the past. We proved that lookback is more commonly observed than lookahead, and therefore the lookback-based protocol has a broader application range. Furthermore, the lookback-based protocol can make the simulation run faster than the speed limit imposed on the lookahead-based protocols by the accumulated event times on the critical path.

Finally, we will close the tutorial classifying lookback into four types: direct strong lookback, universal strong lookback, direct weak lookback, and universal weak lookback. They are defined based on the concepts of absolute impact time and dynamic impact time. We discuss their relationships by considering for each type of lookback whether rollbacks or anti-messages are avoided. From different types of lookback, we will also describe three optimization techniques for optimistic simulation and point out their advantages over lazy cancellation. Finally, we will show that all four types of lookback exist in the PCS network simulation and can be exploited by either lookback-based protocols or optimistic protocols. We will also present application of the lookback-based protocol to Closed Queuing Network simulation, which produced promising speedup on multiprocessors insensitive to workloads. We will also describe how lookback-enabled components can be mixed with ordinary components, and will demonstrate that such a mix can attain even better performance than pure lookback implementation when there is a high density of workloads. This approach of mixing different components can also significantly reduce the complexity of simulation modeling.

Biographical notes of the speakers:

Dr. Boleslaw K. Szymanski is a Professor at the Department of Computer Science and a member of the Scientific Computation Research Center, Rensselaer Polytechnic Institute. He received his Ph.D. in Computer Science from National Academy of Sciences in Warsaw, Poland, in 1976. He was a post-doctoral fellow at the Aberdeen University in Scotland and on the faculty of the Department of Computer and Information Sciences at University of Pennsylvania. He is an author and co-author of more than hundred scientific publications and an editor of three books. Dr. Szymanski is also on Editor-in-Chief of Scientific Programming and on the editorial boards of other journals. Dr. Szymanski is an IEEE fellow and a member of the IEEE Computer Society, and Association for Computing Machinery where he was a National Lecturer.

Gilbert Chen is a graduate student at the Department of Computer Science, Rensselaer Polytechnic Institute. He received an MS and a BS in electronic engineering from Tsinghua University, P.R. China. His research interests include parallel discrete event simulation, component-based software development. His Ph.D. thesis, nearly complete, introduces lookback and investigates its properties and applications.