Load-Aware Dynamic Time Synchronization in Parallel Discrete Event Simulation

Ali Eker, Yehia Arafa, Abdel-Hameed A. Badawy, Nandakishore Santhi, Stephan Eidenbenz, Dmitry Ponomarev

ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (SIGSIM-PADS)
June 1st 2021
A New Time Synchronization Schema: Hybrid PDES

Hybrid PDES: 3X Performance Improvement

- Conservative Synchronization
- Optimistic Synchronization
Presentation Outline

1. Background
2. Motivation
3. Hybrid PDES
4. Experimental Results
Background

- Parallel Discrete Event Simulation (PDES) for high-performance, high-fidelity simulations

- Logical Processes (LPs)

- Time-stamped Event Messages
Causality Order between Event Messages must be Preserved for a Consistent Simulation!

Conservative Synchronization
- Time-Window Computation
- Strong Consistency

Optimistic Synchronization
- Time Warp Protocol [1,2]
- Eventual Consistency

Critical Observation

Balanced Load Distribution

Imbalanced Load Distribution
Hypothesis

Conservative/Optimistic Synchronization is suboptimal if the load distribution changes dynamically:

- Traffic Simulations
- Network Simulations
- Performance Prediction
Hybrid PDES

- Induce the load distribution.
- Switch between Conservative and Optimistic.
Objective

- Exploit the dynamically changing load distributions.
- Run Optimistically as long as it is beneficial for the system performance.
Hybrid PDES Overview

- Process 1
- Process 2
- Process 3
- Process 4

Wall-Clock Time

Conservative Synchronization
Optimistic Synchronization

Balanced
Imbalanced Message Distribution
How/When to Switch?

Metrics computed periodically to induce the load distribution:

Sigma Value

Efficiency of Simulation
Simulation Metric 1

Sigma Value

- Variance among processes’ remote message receipts.
- Triggers a switch from Conservative to Optimistic mode.
Efficiency of Simulation

- Total number of events committed over executed.
- Triggers a switch from Optimistic to Conservative mode.
Experimental Setup

Simian PDES Framework [3]:

• SimianLua
• SimianPie

Applications

- Phold Model
- Los Alamos-PDES Benchmark Suite
- Performance Prediction Tool
- PPT-GPU [4]

Hardware Platforms

- Intel’s Second Gen. Xeon Phi: Knights Landing
- AMD’s EPYC
Experimental Results

1. Phold Based Models in SimianLua
2. La-pdes Based Models in SimianLua
3. La-pdes Based Models in SimianPie
4. PPT-GPU in SimianPie
1. Phold Based Models in Lua

Balanced Load Distribution

Imbalanced Load Distribution

Interleaved Load Distribution
2. La-pdes Based Models in Lua

Balanced Load Distribution

Imbalanced Load Distribution

Interleaved Load Distribution
3. La-pdes Based Models in Python

Interleaved Load Distribution
4. PPT-GPU in Python

Interleaved Load Distribution
**Conclusion**

**Observation:** Monolithic synchronization is suboptimal.

**Mitigation:** Dynamically switch to the better performing synchronization mechanism.

**Proposal:** Hybrid PDES for 3X performance improvement.
Thank You!

Questions?