Parallel Discrete Event Simulation: A Pedestrian View

Daniel Topa
daniel.topa@hii.com

Huntington Ingalls Industries
Mission Technologies

December 24, 2024

Outline I

1. PDES: Why
2. PDES: What
3. Toy Problems
4. Building and Scaling
5. HPC and PDE

Relevance for SDA

1. Define Parallel Discrete Event Simulation
2. Space domain application
3. Parallelism challenges and opportunities

Approaches

I am slide

Stathopoulos: Personal Web Page [6]

Parallel Discrete Event Simulation

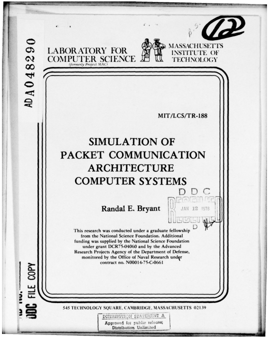
PDES: the execution of a single DES program on a parallel computer

Why PDES?

large simulations consume enormous amounts of time on sequential machines

- engineering
- computer science
- economics
- military applications

Bryant: Simulation of Packet Communication Architecture Systems [1]


ADA048290

LABORATORY FOR COMPUTER SCIENCE (formerly Project MAC)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MIT/LCS/TR-188

SIMULATION OF PACKET COMMUNICATION ARCHITECTURE COMPUTER SYSTEMS

Randal E. Bryant

D D C RECEIVED JAN 12 1978 D

This research was conducted under a graduate fellowship from the National Science Foundation. Additional funding was supplied by the National Science Foundation under grant DCR75-04060 and by the Advanced Research Projects Agency of the Department of Defense, monitored by the Office of Naval Research under contract no. N00014-75-C-0661

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DDC FILE COPY

Chandy & Misra: Simulation of Packet Communication Architecture Systems [2]

440 IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-5, NO. 5, SEPTEMBER 1979

Distributed Simulation: A Case Study in Design and Verification of Distributed Programs

K. MANI CHANDY AND JAYADEV MISRA, MEMBER, IEEE

Abstract—The problem of system simulation is typically solved in a sequential manner due to the wide and intensive sharing of variables by all parts of the system. We propose a distributed solution where processes communicate only through messages with their neighbors; there are no shared variables and there is no central process for message routing or process scheduling. Deadlock is avoided in this system despite the absence of global control. Each process in the solution requires only a limited amount of memory. The correctness of a distributed system is proven by proving the correctness of each of its component processes and then using inductive arguments. The proposed solution has been empirically found to be efficient in preliminary studies. The paper presents formal, detailed proofs of correctness.

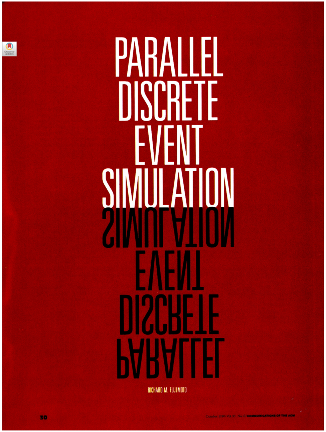
Index Terms—Concurrent processes, distributed systems, performance, program proving, simulation.

by the process up to t. The contents of the message sent out by a process at t, if any, also depend only on the messages received by that process up to t. Note that the output messages of a process may depend both on the content and on the times at which messages were received by the process. Examples of such systems are job shops (where messages are jobs), data base systems, and communications networks.

The Logical System (Simulator)

Each process in the physical system is simulated by a separate logical process. We use the term LP for logical process and PP for physical process. The logic of an LP depends only upon the PP that it is simulating; it is independent of the rest

Fujimoto: Parallel discrete event simulation [3]



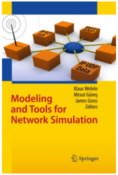
Fujimoto Citations: Military Engagements

1. Frederick Wieland et al. "The performance of a distributed combat simulation with the time warp operating system". In: Concurrency: Practice and Experience 1.1 (1989), pp. 35–50
2. John B Gilmer Jr. "An assessment of 'Time Warp' parallel discrete event simulation algorithm performance.". In: SCS Multiconference on Distributed Simulation. 1988, pp. 45–49

Kunz: Modeling and tools for network simulation [5]

8. Parallel Discrete Event Simulation

Georg Kunz (RWTH Aachen University)



8.1 Introduction

Ever since discrete event simulation has been adopted by a large research community, simulation developers have attempted to draw benefits from executing a simulation on multiple processing units in parallel. Hence, a wide range of research has been conducted on Parallel Discrete Event Simulation (PDES). In this chapter we give an overview of the challenges and approaches of *parallel simulation*. Furthermore, we present a survey of the parallelization capabilities of the network simulators OMNeT++, ns-2, DSIM and JiST.

Experiment

I am slide

Essential Background Knowledge

1. Conservative vs. optimistic mechanisms
2. Deadlock management strategies
3. Parallelism challenges and opportunities

Approaches

I am slide

Deadlock Management

I am slide

Parallelism: Problems and Promise

I am slide

Essential Background Knowledge

1. MM1 Queue Simulation
2. Traffic flow
3. Epidemic modeling
4. Predator-prey dynamics
5. Scripts: Python, Julia, Octave

Approaches

I am slide

Deadlock Management

I am slide

SIR models with discrete events

I am slide

Predator-prey dynamics

I am slide

From Toy Models

1. Conservative vs. optimistic mechanisms
2. Deadlock management strategies
3. Parallelism challenges and opportunities

Libraries

1. **Adevs**
2. **BigSim**
3. **JiST**

Deadlock Management

1. Adevs
2. BigSim
3. JiST

Parallelism: Problems and Promise

1. **NVIDIA**
2. **TAU**
3. **Vampir**

Essential Background Knowledge

1. Benefits of distributed and parallel systems
2. HPC pipelines: MPI or OpenMP
3. HPC pipelines: Coarrays
4. HPC workflows

Approaches

I am slide

Message Passing Interface: MPI

I am slide

OpenMP

I am slide

Coarrays

I am slide

Bibliography I

[1] R. E. Bryant. Simulation of packet communication architecture computer systems. Tech. rep. USA, 1977.

[2] K. Mani Chandy and Jayadev Misra. "Distributed simulation: A case study in design and verification of distributed programs". In: IEEE Transactions on software engineering 5 (1979), pp. 440–452.

[3] Richard M Fujimoto. "Parallel discrete event simulation". In: Communications of the ACM 33.10 (1990), pp. 30–53.

Bibliography II

[4] John B Gilmer Jr. "An assessment of 'Time Warp' parallel discrete event simulation algorithm performance.". In: SCS Multiconference on Distributed Simulation. 1988, pp. 45–49.

[5] Georg Kunz. "Parallel discrete event simulation". In: Modeling and Tools for Network Simulation. Springer, 2010, pp. 121–131.

[6] Andreas Stathopoulos. Parallel discrete event simulation.

Bibliography III

[7] Frederick Wieland et al. "The performance of a distributed combat simulation with the time warp operating system". In: Concurrency: Practice and Experience 1.1 (1989), pp. 35–50.

Parallel Discrete Event Simulation: A Pedestrian View

Daniel Topa
daniel.topa@hii.com

Huntington Ingalls Industries
Mission Technologies

December 24, 2024