Parallel Discrete Event Simulation: A Pedestrian View

Daniel Topa
daniel.topa@hii.com

Huntington Ingalls Industries
Mission Technologies

December 23, 2024

Outline I

1. About PDES
2. Literature Sampling
3. Parallel Computation Tools
4. Sample PDES

Main Points

1. Naïve expansion
2. Dedicated Web Sites
3. Federation of American Scientists

Bertrand's Paper En français

Concepts I

1. Conservative vs. Optimistic Mechanisms
2. Deadlock Avoidance
3. Deadlock Detection and Recovery
4. Synchronous Operation
5. Conditional Knowledge
6. Lazy Cancellation
7. Lazy Reevaluation
8. Direct Cancellation

A. Gupta: Northwestern University

Writing a Discrete Event Simulation: ten easy lessons

Proof Types

1. Academic papers
2. Books
3. Unpublished

Parallel Discrete Event Simulation [2]

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.

Parallel Discrete Event Simulation[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

[6]

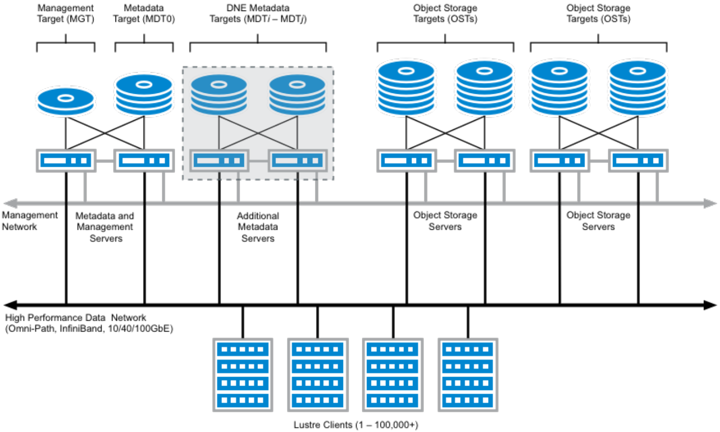
Proof Types

1. File systems (Lustre)
2. Performance Analysis Tools (Vampir, Tau, NVIDIA)
3. Data formats (HDF)
4. Parallel databases

Lustre File System

Open-source, parallel file system

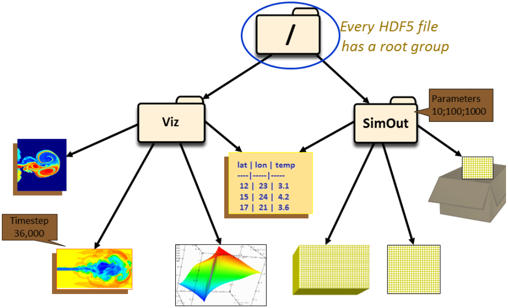
Lustre File System Wiki



Hierarchical Data Format (HDF)

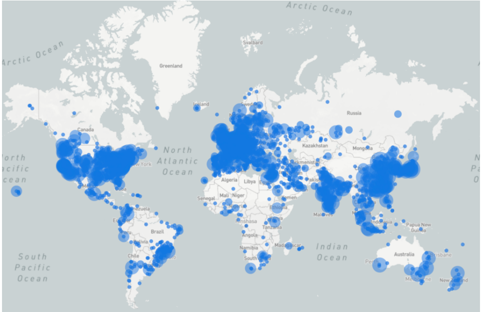
Set of file formats designed to store and organize large amounts of data

Hierarchical Data Format (HDF)



Sharing data

Hierarchical Data Format (HDF) Weekly Clinic



Tuesday, 11:20 Mountain

NVIDIA Performance Analysis Tools

NVIDIA Nsight Systems

NVIDIA® Nsight™ Systems is a system-wide performance analysis tool designed to visualize application's algorithm, help you select the largest opportunities to optimize, and tune to scale efficiently across any quantity of CPUs and GPUs in your computer; from laptops to DGX servers.

NVIDIA® Nsight™

The ultimate development platform for heterogeneous computing. Work with powerful debugging and profiling tools, optimize the performance of your CPU and GPU code. Find out about the Eclipse Edition and the graphics debugging enabled Visual Studio Edition.

NVIDIA Visual Profiler

This is a cross-platform performance profiling tool that delivers developers vital feedback for optimizing CUDA C/C++ applications. First introduced in 2008, Visual Profiler supports all CUDA capable NVIDIA GPUs shipped since 2006 on Linux, Mac OS X, and Windows.

TAU Performance System®

This is a profiling and tracing toolkit for performance analysis of hybrid parallel programs written in CUDA, and pyCUDA., and OpenACC.

VampirTrace

A performance monitor which comes with CUDA, and PyCUDA support to give detailed insight into the runtime behavior of accelerators. Enables extensive performance analysis and optimization of hybrid programs.

The PAPI CUDA Component

A hardware performance counter measurement technology for the NVIDIA CUDA platform which provides access to the hardware counters inside the GPU. Provides detailed performance counter information regarding the execution of GPU kernels.

The NVIDIA CUDA Profiling Tools Interface

(CUPTI) provides performance analysis tools with detailed information about GPU usage in a system. CUPTI is used by performance analysis tools such as the NVIDIA Visual Profiler, TAU and Vampir Trace.

NVIDIA Topology-Aware GPU Selection

(NVTAGS) is a toolset for HPC applications that enables faster solve times with high GPU communication-to-application run-time ratios. NVTAGS intelligently and automatically assigns GPUs to message passing interface (MPI) processes, thereby reducing overall GPU-to-GPU communication time.

Vampir

Display and analyze arbitrary program behavior
at any level of detail

TAU: Tuning and Analysis Utilities

Portable profiling and tracing toolkit for performance analysis of parallel programs written in Fortran, C, C++, UPC, Java, Python

TAU: Tuning and Analysis Utilities

1. Sameer S Shende and Allen D Malony. "The TAU parallel performance system". In: The International Journal of High Performance Computing Applications 20.2 (2006), pp. 287–311
2. Kathleen A Lindlan et al. "A tool framework for static and dynamic analysis of object-oriented software with templates". In: SC'00: Proceedings of the 2000 ACM/IEEE Conference on Supercomputing. IEEE. 2000, pp. 49–49

TAU: Tuning and Analysis Utilities

3. A Malony et al. "Performance technology for parallel and distributed component software". In: Concurrency and Computation: Practice and Experience 17.2-4 (2005), pp. 117–141
4. David E Bernholdt et al. "A component architecture for high-performance scientific computing". In: The International Journal of High Performance Computing Applications 20.2 (2006), pp. 163–202

Thrust

1. GitHub– Thrust: The C++ Parallel Algorithms Library
2. GitHub– About CUB
3. NVIDIA developer blog: Popular Open Source Thrust and CUB Libraries Updated
4. Introduction to CUDA Libraries : Thrust

Tools

1. Naïve expansion
2. Dedicated Web Sites
3. Federation of American Scientists

Tools

1. Adevs (C++ library)
2. Dedicated Web Sites
3. Federation of American Scientists

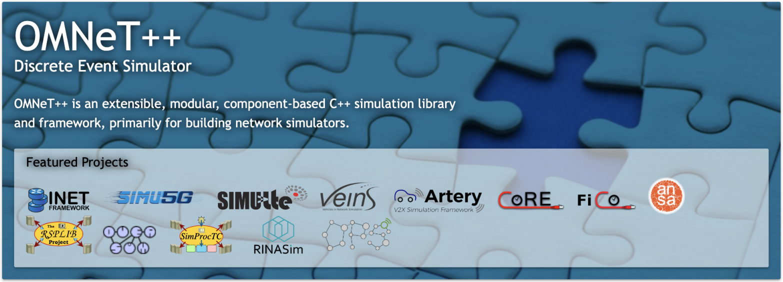
Adevs

1. C++ library for building simulations that are
   - discrete event
   - mixed discrete event & continuous
2. Foundation formulations
   - Discrete Event System Specification (DEVS)
   - Dynamic DEVS
3. Supports runtime system for OpenModelica

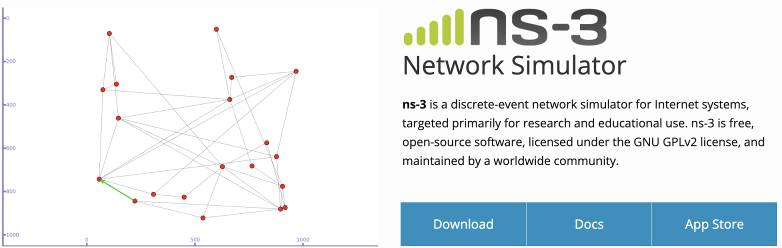
Network Parallel Discrete Event Simulations

1. OMNet
2. ns-3
3. DSIM
4. JiST

OMNET++


OMNeT++
Discrete Event Simulator
OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators.
Featured Projects
INET FRAMEWORK
SIMU5G
SIMUlte
veins
Artery
V2X Simulation Framework
CoRE
FiCo
an sa
The RSPLIB Project
SimProcTC
RINASim

ns-3


0
200
400
600
800
1000
0
500
1000
ns-3
Network Simulator
ns-3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. ns-3 is free, open-source software, licensed under the GNU GPLv2 license, and maintained by a worldwide community.
Download
Docs
App Store

DSIM

Open Simulation Solutions

for Research, Development and Training

D-SIM and D-WORLD are the perfect software solutions for Open Simulation in which you control all the simulators in one distributed virtual world

D-SIM

Your solution for rapid development in an open simulation framework.

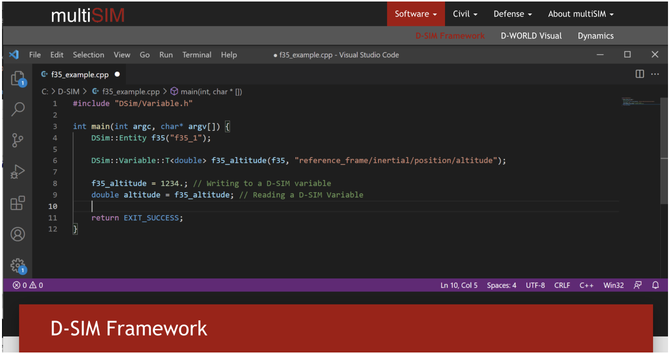
D-WORLD

Your virtual reality render engine for realistic simulations in one distributed world.

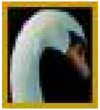
Dynamics

Your high-fidelity physics solution for aircraft and vehicle simulations.

DSIM


multiSIM
Software
Civil
Defense
About multiSIM
D-SIM Framework
D-WORLD Visual
Dynamics
File Edit Selection View Go Run Terminal Help
• f35_example.cpp - Visual Studio Code
f35_example.cpp
C: > D-SIM > f35_example.cpp > main(int, char * [])
#include "DSim/Variable.h"
int main(int argc, char* argv[]) {
DSim::Entity f35("f35_1");
DSim::Variable::T<double> f35_altitude(f35, "reference_frame/inertial/position/altitude");
f35_altitude = 1234.; // Writing to a D-SIM variable
double altitude = f35_altitude; // Reading a D-SIM Variable
return EXIT_SUCCESS;
}
Ln 10, Col 5 Spaces: 4 UTF-8 CRLF C++ Win32
D-SIM Framework

JiST

Java in Simulation Time

Scalable Wireless Ad hoc Network Simulator

Bibliography I

*

[1] David E Bernholdt et al. “A component architecture for high-performance scientific computing”. In: The International Journal of High Performance Computing Applications 20.2 (2006), pp. 163–202.

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

Bibliography II

[3] Kathleen A Lindlan et al. “A tool framework for static and dynamic analysis of object-oriented software with templates”. In: SC’00: Proceedings of the 2000 ACM/IEEE Conference on Supercomputing. IEEE. 2000, pp. 49–49.

[4] A Malony et al. “Performance technology for parallel and distributed component software”. In: Concurrency and Computation: Practice and Experience 17.2-4 (2005), pp. 117–141.

Bibliography III

[5] Sameer S Shende and Allen D Malony. "The TAU parallel performance system". In: The International Journal of High Performance Computing Applications 20.2 (2006), pp. 287–311.

[6] Andreas Stathopoulos. Parallel discrete event simulation.

Parallel Discrete Event Simulation: A Pedestrian View

Daniel Topa
daniel.topa@hii.com

Huntington Ingalls Industries
Mission Technologies

December 23, 2024