Mission Planning Simulation and Design Software Scaling for Shared and Distributed Memory Computing

Sam Siewert
California State University
400 W. First St., Chico, CA 95929-0410
sbsiewert@csuchico.edu

Angel Martinez-Sanchez
University of California Merced
5200 Lake Rd, Merced, CA 95343
amartinez-sanchez2@ucmerced.edu

Jacob Collins
California State University
jbcollins@csuchico.edu

Chirag Dhamange
California State University
cdhamange@csuchico.edu

Stevie D. Littleton Jr.
California State University
sdlittleton@csuchico.edu

***Abstract*— Mission planning and simulation software architecture should ideally provide an open, scalable solution for a wide range of aerospace systems and allow for analysis and design optimization. Existing open-source mission planning and simulation tools such as GMAT (General Mission Analysis Tool), Nyx-space, Godot, ODtbx, and NOS3 provide propagators for space systems celestial mechanics and trajectory determination along with analysis and design tools. However, they all could be improved with parallel scaling software design patterns for propagation and optimization features. Specifically, this paper explores the use of the most current object oriented, shared, and distributed memory parallel programming methods for both scale-up and scale-out architectures. Further, the paper presents a software design pattern for use by the core simulation and optimization features of mission planning by providing a simple stand-alone example along with work to create a new prototype extension to existing open source.**

Many mission planning, simulation and optimization objectives benefit from open-source parallel scaling, especially with Monte Carlo optimization, and problem scaling to simulate many celestial bodies and space vehicles, over long durations. Present tools either lack parallel scaling or use proprietary methods, often limited to one type of parallelism (e.g., shared memory). The goal of this investigation is to re-engineer the core propagation and Monte Carlo features for speed-up and scaling with portable parallel software engineering design patterns that integrate both shared memory and distributed memory parallel methods. While co-processing can also be used in addition, based upon lack of open solutions for co-processors such as GP-GPU (General Purpose – Graphic Processing Units), proprietary methods such as "CUDA" (Compute Unified Device Architecture), Advance Micro Devices "ROCm", and Apple "Metal" are avoided in this study to avoid incorporating proprietary parallel software. Longer term, open co-processing such as OpenCL can be investigated, but we have found that a focus on open shared and distributed memory scaling provides significant advancement.

In this paper, we examine and evaluate the relative merits of combining OpenMP (shared memory) and MPI (distributed memory) to extend existing mission planning and simulation tools. Specifically, the tools benefit from scaling for Monte Carlo analysis and multi-body simulation supported by current supercomputing machines which have millions of distributed memory cores. Various methods to provide parallel features for simulation, such as Intel TBB (Thread Building Blocks) extensions to GMAT, are a positive step toward open-source simulation scaling. However, at the same time, there has been full replacement of open-source tools like GMAT with proprietary tools like ANSYS System Toolkit for mission design. Preserving open-source mission design and analysis tools that are competitive in terms of performance, scaling, and features is beneficial to the broader space science community by providing full source options that are portable and extensible. The open-source design pattern will be presented by example in a small-scale simulation along with experience using this pattern in existing larger open-source mission design and analysis tools. Results showing comparison of runtimes for sequential compared to parallel as well as existing proprietary threaded runs are presented and analyzed in this paper.

Overall, we identified five major parallel design patterns for simulation and found that four of the five work well and are widely used, but not comprehensively, and the fifth we proposed has specific challenge for use in MPAD simulations.

979-8-3315-7360-7/26/$31.00 ©2026 IEEE