Performance and Accuracy of WARP - A Framework for Continuous Energy Monte Carlo Neutron Transport in General 3D Geometries on GPUs

Ryan M. Bergmann*, Kelly Rowland, Nikola Radnović, Jasmina L. Vujić

Department of Nuclear Engineering, 4155 Etcheverry Hall, University of California - Berkeley, Berkeley, CA 94703-1730

Abstract

In this companion paper to "Algorithmic Choices in WARP - A Framework for Continuous Energy Monte Carlo Neutron Transport in General 3D Geometries on GPUs" (doi:10.1016/j.anucene.2014.10.039), the WARP Monte Carlo neutron transport framework for GPUs is benchmarked against production-level CPU Monte Carlo neutron transport codes for both performance and accuracy. Fission source distributions, flux spectra, and multiplication factors calculated by WARP are compared to those from Serpent v2.XX.X and MCNP v6.1. for identical materials and geometries. Runtimes are also reported.

Keywords: Monte Carlo, Neutron Transport, GPU, CUDA, CUDPP, OptiX

1. Introduction

Developing WARP was motivated by modern supercomputers commonly being built with graphics processing units (GPU) coprocessor cards in their nodes to increase their computational efficiency and performance []. Compared to more common central processing units, or CPUs, GPUs have a larger aggregate memory bandwidth, much larger rate of floating-point operations

*Corresponding author. Tel.: +41.76.687.53.09.

Email addresses: ryanmbergmann@gmail.com (Ryan M. Bergmann), krowland@berkeley.edu (Kelly Rowland), radnovicn@gmail.com (Nikola Radnović), vujic@nuc.berkeley.edu (Jasmina L. Vujić)

per second (FLOPS), and lower energy consumption per FLOP[]. GPUs execute efficiently on data-parallel problems [], and since most CPU codes are task-parallel, the algorithms used had to be reconsidered. Data-parallelism is simply parallelism that arrises from operating on many different pieces of data at one time, whereas task-parallelism is parallelism that arises from running many concurrent tasks at one time which act on a single piece of data. Figure 1 shows and illustration of the difference between a data-parallel and a task-parallel neutron transport loop.

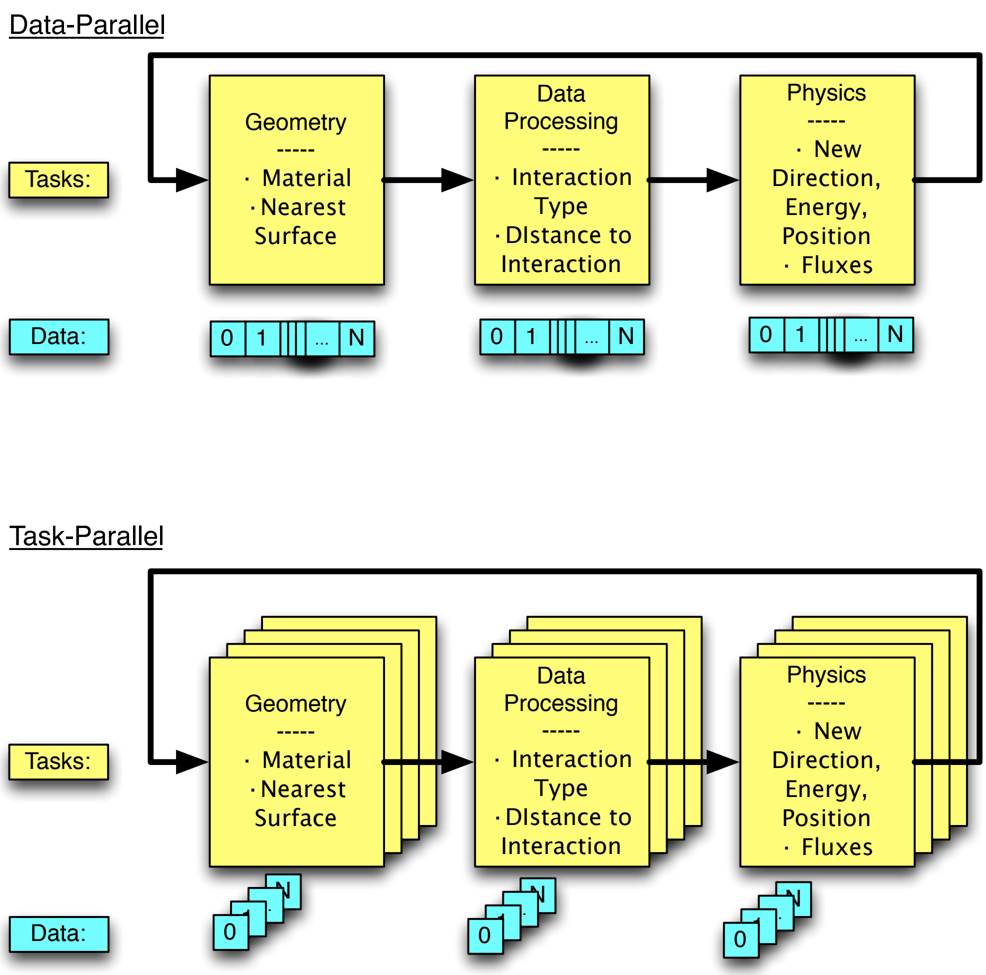

Figure 1: Data-parallel neutron trasport loop vs. a task-parallel transport loop for N neutrons in parallel.

Execution on GPUs also requires additional data management, since the on-chip memory of the GPU is separate from the host CPUs memory []. Execution on NVIDIA GPUs also required code to be written in CUDA, which is basically a set of extensions for C/C++. The simplest way to accommodate all these requirements was to write a new code from scratch, and ultimately resulted in WARP.

In this paper, results calculated by WARP are compared against those calculated by Serpent 2.XX.X and MCNP 6.1, two widely-used production-level Monte Carlo neutron transport codes, in order to ensure the accuracy of WARP and to highlight its performance differences. The details about the algorithms used in WARP are discussed in [**?**].

2. Features of WARP

2.1. Physics

limitations too

2.2. Geometry

limitations too

2.3. Interface

limitations too

3. Tests

3.1. Test 1

3.2. Test 2

3.3. Test 3

3.4. Test 4

4. Results

4.1. Test 1

4.2. Test 2

4.3. Test 3

4.4. Test 4

4.5. Summary

5. Conclusions and Future Development

Acknowledgements

This research is based upon work partially supported by the U.S. Department of Energy National Nuclear Security Administration under Award Number DENA0000979 through the Nuclear Science and Security Consortium: http://nssc.berkeley.edu.