GPU Acceleration of History-Based Multigroup Monte Carlo

Achievement: Developed scalable threaded algorithms for Monte Carlo neutron transport on the GPU. Work awarded Mathematics & Computation Division Best Summary+Presentation Award at ANS 2016 Winter Meeting

Significance and Impact: Understanding how traditional Monte Carlo neutron transport algorithms map to multiple GPU architectures and how algorithmic variants can improve performance.

Research Details:
- GPU implementation of history based multigroup Monte Carlo neutron transport.
- Reported results efficiently using multiple GPUs allowing for execution on Titan.
- Algorithmic improvements resulting in single GPU speedup of 2–4x over 8-core shared memory CPU implementation.

Sponsor/Facility: This work was supported by the ORNL LDRD program. This manuscript has been authored by UT- Battelle, LLC, under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

PI and affiliation: Steven Hamilton – Reactor and Nuclear Systems Division, Oak Ridge National Laboratory

Team: Steven Hamilton (RNSD), Thomas Evans (RNSD), Stuart Slattery (CSMD)


Overview: Accurate solutions of the radiation transport equation play an important role in the analysis of many nuclear systems. Because there is no need to discretize the problem, Monte Carlo methods offer the most accurate possible solutions to the transport equation. However, significant computational effort is required to reduce the stochastic noise inherent to Monte Carlo methods. Current trends in high performance computing are moving towards vectorized, single instruction multiple data (SIMD), architectures such as Intel Xeon Phis and graphics processing units (GPUs) due to their favorable ratio of performance to power consumption. The challenge of adapting Monte Carlo transport to vectorized computing architectures was first addressed by Brown with an algorithm known as event-based Monte Carlo. In this approach, rather than simulating individual particle histories from birth to death, groups of particles are processed one event at a time. This allows each event batch to perform the same operations in lockstep, enabling use of the vector hardware. Although modern vectorized hardware is more versatile than the early vector machines, reducing so-called thread divergence is nonetheless still an important consideration for algorithm design. Many of the algorithms proposed so far for Monte Carlo transport on
advanced architectures have focused on variants of event-based transport. Some evidence, however, suggests that low-level thread divergence, resulting from short-lived branching statements, may not be detrimental to overall performance on GPUs if the effect on achievable memory bandwidth is considered. This paper characterizes the performance of a traditional history-based Monte Carlo algorithm on GPUs. We show that with only minor modifications to traditional algorithms, significant performance gains relative to CPUs are possible.