Procedia Computer Science

Volume 80, 2016, Pages 484–495

ICCS 2016. The International Conference on Computational Science

Competing Energy Lookup Algorithms in Monte Carlo Neutron Transport Calculations and Their Optimization on CPU and Intel MIC Architectures

Yunsong Wang[1], Emeric Brun[2], Fausto Malvagi[2], and Christophe Calvin[3]

[1] Maison de la Simulation, CEA, CNRS, Univ. Paris-Sud, UVSQ, Univ. Paris-Saclay
yunsong.wang@cea.fr
[2] CEA/DEN/DANS/DM2S/SERMA
[3] CEA/DRF/D3P
CEA Saclay, F-91191, Gif-sur-Yvette, France

Abstract

The Monte Carlo method is a common and accurate way to model neutron transport with minimal approximations. However, such method is rather time-consuming due to its slow convergence rate. More specifically, the energy lookup process for cross sections can take up to 80% of overall computing time and therefore becomes an important performance hotspot. Several optimization solutions have been already proposed: unionized grid, hashing and fractional cascading methods. In this paper we revisit those algorithms for both CPU and manycore (Intel MIC) architectures and introduce vectorized versions. Tests are performed with the PATMOS Monte Carlo prototype, and algorithms are evaluated and compared in terms of time performance and memory usage. Results show that significant speedup can be achieved over the conventional binary search on both CPU and Intel MIC. Further optimization with vectorization instructions has been proved very efficient on Intel MIC architecture due to its 512-bit Vector Processing Unit (VPU); on CPU this improvement is limited by the smaller VPU width.

Keywords: Monte Carlo, neutron transport, cross section, table lookup, Intel MIC, vectorization

1 Introduction

Monte Carlo (MC) transport simulations are widely used in the nuclear community to perform reference calculations. This method simulates the physics by following a neutron in its travels inside a system from birth to absorption or leakage. This random walk is governed by interaction probabilities described by microscopic cross sections. Macroscopic quantities like neutron densities can be estimated from large samples of histories, which makes the MC method more computationally expensive than other approaches. The advantage of the MC method is that

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

doi:10.1016/j.procs.2016.05.327

since the individual histories are independent, it is an ideal candidate for parallel computing. On the other hand, each history is different and there is little evidence of natural vectorization.

Several difficulties have been identified when porting MC codes to modern architectures: hybrid parallelism, memory sharing, cache misses, limited use of vector processing units. Modern computing accelerators evolve constantly and have distinct strategies which thus require specific optimization efforts.

In order to explore both the challenges and the benefits brought by emerging accelerators for the MC codes, a new MC neutron transport prototype called PATMOS (PArticle Transport Monte Carlo Object-oriented System) is currently under development at CEA [1]. PATMOS is lightweight and portable, but also complex enough to represent a real simulation. Its goal is to test competing algorithms on a variety of hardware so as to explore their computing potentials and get the best compromise between performance and sustainability of the code. This prototype is entirely written in C++ standard 11/14 [2]. PATMOS implements a hybrid parallelism based on MPI and OpenMP or standard C++ threads. In a multi-threaded environment, the total number of particles to be simulated is evenly dispatched to all the available threads.

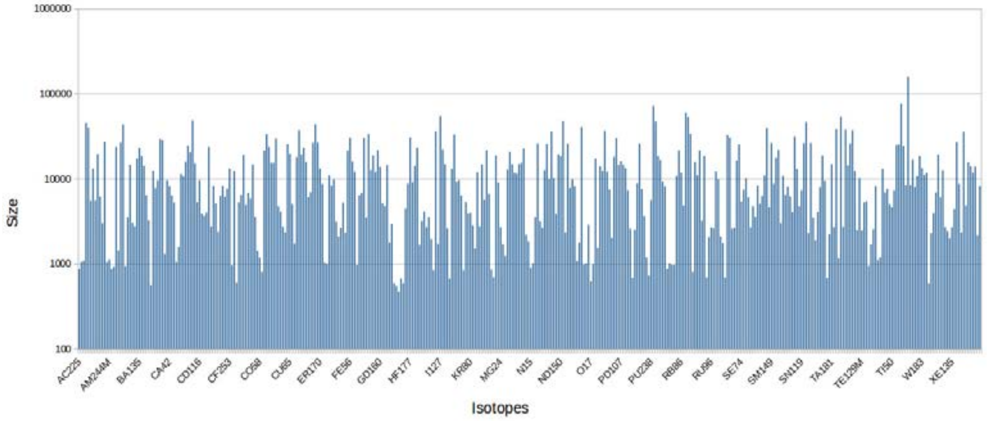

Figure 1: Isotopic energy table lengths for 390 isotopes at $T = 300$K. The minimum is for $^{3}\mathtt{H}$, which has only 469 energy points, and the largest is for $^{238}\mathtt{U}$ with 156,976 points. The average length over all isotopes is around 12,000.

In MC transport calculations, the cross sections represent the interaction probabilities of the particle with the underlying medium. Isotopic cross sections, which depend on the kinetic energy of the incident particle, are stored in tables of up to 150,000 couples (E_i, σ_i) meant for linear-linear interpolation (Figure 1). These isotopic tables typically have between 10^2 and 10^5 values, depending on the variations of the cross-sections, number of resonances, threshold reactions, etc [3].

Every time a particle with energy E suffers a collision or crosses a material interface, the material total cross section $\Sigma(E)$ needs to be recalculated as the sum of all isotopic cross sections $\sigma_i(E)$ times the isotopic concentrations N_i. That is $\Sigma(E) = \sum_i N_i \sigma_i(E)$, where the isotopic cross sections are computed as:

$$\sigma_i(E) = \sigma_i(E_k) + \frac{E - E_k}{E_{k+1} - E_k}[\sigma_i(E_{k+1}) - \sigma_i(E_k)]$$

where E_k and E_{k+1} are the nearest lower and upper energy points in the isotopic energy grid which bound the random energy E.