Contents lists available at ScienceDirect

Parallel Computing

journal homepage: www.elsevier.com/locate/parco

Monte Carlo domain decomposition for robust nuclear reactor analysis

Nicholas Horelik [a,*], Andrew Siegel [b], Benoit Forget [a], Kord Smith [a]

[a] Massachusetts Institute of Technology, Department of Nuclear Science and Engineering, 77 Massachusetts Avenue, Building 24, Cambridge, MA 02139, United States

[b] Argonne National Laboratory, Mathematics and Computational Sciences (MCS) Division, 9700 Cass Avenue, Argonne, IL 60439, United States

ARTICLE INFO

Article history:
Received 4 April 2014
Received in revised form 12 August 2014
Accepted 9 October 2014
Available online 20 October 2014

Keywords:
Monte Carlo
Domain decomposition
Load balancing
Neutron transport
Nuclear reactor analyses

ABSTRACT

Monte Carlo (MC) neutral particle transport codes are considered the *gold-standard* for nuclear simulations, but they cannot be robustly applied to high-fidelity nuclear reactor analysis without accommodating several terabytes of materials and tally data. While this is not a large amount of aggregate data for a typical high performance computer, MC methods are only embarrassingly parallel when the key data structures are replicated for each processing element, an approach which is likely infeasible on future machines. The present work explores the use of spatial domain decomposition to make full-scale nuclear reactor simulations tractable with Monte Carlo methods, presenting a simple implementation in a production-scale code. Good performance is achieved for mesh-tallies of up to 2.39 TB distributed across 512 compute nodes while running a full-core reactor benchmark on the Mira Blue Gene/Q supercomputer at the Argonne National Laboratory. In addition, the effects of load imbalances are explored with an updated performance model that is empirically validated against observed timing results. Several load balancing techniques are also implemented to demonstrate that imbalances can be largely mitigated, including a new and efficient way to distribute extra compute resources across finer domain meshes.

1. Introduction

Next-generation high-performance-computing (HPC) architectures will increasingly utilize on-node parallelism to achieve improvements in peak FLOP rates at improved power efficiency [1–3]. For many applications the increased processor performance will have a significant impact on the fidelity of the physical models, potentially enabling the simulation of a much broader range of physical phenomenon for significantly longer timescales or at much higher resolution. However, at the same time the total memory is growing at a slower rate than the aggregate available processing power, so that the amount of memory available to each individual processing unit is descreasing ([4,5]). This new FLOP/memory balance will leave us in a regime quite distinct from what has become familiar over the past twenty years, requiring in many cases non-trivial adaptations of traditional methods in order to take advantage of the increased parallelism.

One prime example can be found in the field of nuclear reactor physics, where stochastic Monte Carlo (MC) particle transport methods have the potential to be a highly accurate, general-purpose tool for robust for full core reactor simulations [6].

[*] Corresponding author.
E-mail addresses: nhorelik@mit.edu (N. Horelik), siegela@mcs.anl.gov (A. Siegel), bforget@mit.edu (B. Forget), kord@mit.edu (K. Smith).

http://dx.doi.org/10.1016/j.parco.2014.10.001
0167-8191/© 2014 Elsevier B.V. All rights reserved.

MC methods can sample directly from evaluated nuclear data to carry out random walks for individual particles through an arbitrarily complex geometry with minimal approximations. They can be used to determine power distributions, dose rates, and other quantities critical to assessing the safety and performance of nuclear systems. However, for certain classes of problems MC methods require immense computational effort to achieve good statistical convergence, making their routine use impractical [7]. Thus, they have traditionally been relegated to a more limited set of applications, such as benchmarking and validation of lower-fidelity methods, or for the analysis of smaller systems with less complex physics. It is desirable to shift this paradigm, especially with the complexity of next-generation reactor designs presenting new challenges to the existing suite of highly-specialized reactor simulation tools.

While the performance improvements of next-generation systems might overcome the time-to-solution limitations of MC methods, a number of algorithmic challenges stand in the way of harnessing the increased processing power – especially in the presence of reduced-memory environments. The root of the problem is that effective parallelization of MC methods has traditionally been carried out by performing parallelization *over particles*, where each processing element tracks a subset of particle histories through (potentially) the entire domain. This approach requires the replication of domain meta-data on each distributed processing element. While this may be a relatively small memory footprint for simple problems, for a robust analysis of power reactors the required data structures are much too large for local memory. We note that this problem is present even on existing leadership class machines, and that a number of simplifications and approximations are made in an attempt to retain the traditional approach and minimize the negative performance impact. On future HPC systems the problem is expected to be exacerbated.

The details of the full-scale reactor problem are discussed in [8]. The basic idea is that a very fine spatial mesh is needed for a robust full-core reactor depletion analysis: several Terabytes of memory are required to hold reaction rate tallies and isotopic abundances for several hundreds of nuclides in each region. For this problem, the authors of [7,8] detail the several well-known performance hurdles that must be overcome before MC codes can be used routinely: (i) excessive overall time to solution for adequate statistical convergence; (ii) inadequate memory for reaction rate tallies and material composition data, and (iii) inadequate memory for temperature-dependent nuclear cross section data. Several recent works aim to address issues (i) and (iii) [9–12], but issue (ii) requires additional attention.

Implementing domain decomposition seems like an obvious solution, but the routine use of this approach is made difficult by the large amount of particle communication that would be required between domains (billions of particles for real problems), and the accompanying potential for significant parallel inefficiencies due to load imbalances [6]. Instead, traditional parallel MC codes employ full domain replication to take advantage of the embarrassingly-parallel nature of neutral particle tracking. With this scheme, all material and tally data is allocated on each distributed computational node [13,14] and synchronized between fission source iterations: a relatively inexpensive operation [15]. Given on-node memory constraints, it is clear that this parallelization scheme is not capable of tackling reactor problems on its own.

One approach for handling this memory limitation without domain decomposition is data decomposition. For example, the method described in [16] stores tally data on a set of distributed "server" processes, to which tracking processes send tally writes via asynchronous MPI sends. Indeed, reasonable performance was observed in [17] for a variety of typical computer parameters with tallies at a scale relevant to full-core analyses. In principle a similar concept can be applied to the treatment of materials data, but it is not clear what effect this will have on particle tracking rates.

On the other hand, domain decomposition immediately solves the data problems for both materials and tallies. Specifically, when memory pertaining to certain regions is allocated only on nodes that track those regions, the footprint on each is reduced inversely to the number of spatial domains used. This is not a new concept for MC codes [18–22], but to our knowledge it has not been applied to full-core 3D reactor analyses. For this class of problem it still needs to be demonstrated that the concept is feasible for realistic calculations that incur the true memory and particle communication burden.

The ability to model and predict the extent of particle communication costs and load imbalances has been investigated in [23,24] for a range of machine and problem parameters with a simplified MC code. In these analyses it is demonstrated that these costs can be computed from the peaking factor and spatial statistics of the problem (i.e., per-domain leakage fractions), and that only modest penalties are predicted for a typical reactor geometry.

These results motivated the present work, where we implement domain decomposition in a full-physics MC code (OpenMC, [14]) and explore the full-size reactor problem for both tally and particle communication performance characteristics.

2. Domain decomposition implementation

2.1. Model

The present work implements domain decomposition by adding particle synchronization *stages* during each generation of particles, as described in [23]. This is much simpler than the algorithms presented in [19–22] but it allows for the derivation of an explicit timing model that can be used to predict how well it will perform for the full-fidelity reactor problem. The changes to the outer loop of the OpenMC eigenvalue iteration routine are shown in lines 4, 5, 9–11, and 15–17 of Algorithm 1.