LA-UR-11-06271

A Monte Carlo Transport mini-App for Exascale Co-Design R&D

Larry J. Cox
Ryan Marcus
XCP Division
Los Alamos National Laboratory

Abstract

Exascale co-design research teams will need representative applications to guide their research. One of the important applications is Monte Carlo Radiation Transport, such as that supported by the ASC software package MCNP. This paper presents a Monte Carlo "mini-App" for Exascale R&D that encompasses many of the necessary algorithmic components of Monte Carlo transport. The physics in kept intentionally simple to avoid export control concerns. However, an effort was also made to make it representative of the "real" physics used in codes such as MCNP.

The primary goal for this MC mini-App is to clearly define the computational requirements and constraints while leaving open as much of the computer science as possible.

A secondary goal of this effort is to develop one or more sample implementations of the App that may help to advance Exascale R&D.

As the Exascale R&D progresses, this MC mini-App will provide a framework to include more realistic physics and to explore emerging hardware/software paradigms.

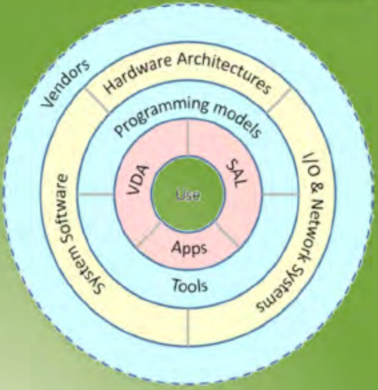
This page not to be included in the electronic poster

Larry Cox, ljcox@lanl.gov, (505) 665-7344
Ryan Marcus, rmarcus@lanl.gov, (505) 500-4365

LA-UR-11-06271

A Monte Carlo Transport mini-App for Exascale Co-Design R&D

Exascale co-design teams require representative applications to guide their R&D. All elements including Use through Vendors will need to utilize these *mini-Apps* to guide development if the final systems are to be of practical use.



One important application domain is ***Monte Carlo radiation transport***, an example of which is the family of codes referred to as MCNP [1]. Monte Carlo (MC) methods are widely used for studying the interaction of radiation with matter in almost every scientific and engineering discipline. These methods are essential to many NNSA missions. Areas of application include radiation shielding, radiography, medical physics, criticality safety, nuclear oil well logging, isotope production, accelerator target design, fission and fusion reactor design, and decontamination and decommissioning of nuclear facilities.

The goal of this effort is to define a neutral-particle (n,g) MC package tailored to solve one or more problems of interest and capable of running on emerging hardware/software paradigms. The *project goals* for the MC mini-App include:

- Clearly defining the *computational* requirements and constraints leaving the *computer* science open [2]
- Developing one or more sample implementations [3]

A Python framework, ***Exa*** [3], is being developed to allow execution of MC algorithmic steps with MPI, Cuda , OpenCL, OpenMP, and/or other paradigms that emerge.

Larry Cox, ljcox@lanl.gov, (505) 665-7344
Ryan Marcus, rmarcus@lanl.gov, (505) 500-4365

LA-UR-11-06271

Monte Carlo mini-App — Exa.py The Python Framework

When developing implementations of "mini-apps" or "co-design applications" for Exascale R&D, it is important to demonstrate an algorithm's potential in several different forms, perhaps including:

- Serial implementations
- Standard parallel implementations (MPI, OpenMP)
- Emerging parallelism frameworks (CUDA, OpenCL)

The goal of the ***Exa*** Python framework [3] is to provide an easy way to implement an algorithm once and run it, near-optimally, on multiple different forms of potential Exascale technologies.

Two basic concepts are utilized in this framework, enabling algorithms written with ***Exa*** to be run on multiple platforms with one source:

–**Array operations: Unary or Binary**

- Operations on each element of an array or arrays resulting in a new array

–**Array reductions:** Reduction to a single element

- Summations, minimums, maximums…

The framework currently has a defined methodology hierarchy:

1. CUDA – if available
2. OpenCL – if available
3. MPI – if available
4. OpenMP – if available
5. Serial execution

The user can also choose the method to use for each algorithmic element.

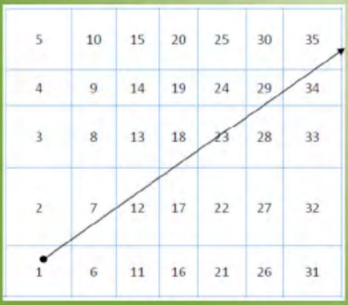
Larry Cox, ljcox@lanl.gov, (505) 665-7344
Ryan Marcus, rmarcus@lanl.gov, (505) 500-4365

LA-UR-11-06271

Monte Carlo mini-App — Prerequisites

An MC transport application requires a variety of information, including:

- **A good random-number generator**
  - A parallel-capable random-number kernel will be needed that provides a predictable, independent random number sequence to an arbitrary number of processes. The method(s) used in MCNP6 will be recast as needed.
- **Initial conditions for a radiation source**
  - A user-defined external source (e.g., radiography)
  - An initial guess at an internal source (e.g., criticality)



- **A description of the transport "universe"**
  - Cartesian, Cylindrical or Spherical axes specified as lists of intercepts
- **Compositional Information**
  - The materials are separately defined and stored
  - The isotopics will remain fixed during the calculation
- **Interaction cross-section data**

$$\lambda(z,E) = \left[\sum_j (z.\rho(j) \times z.m(j).stp(E))\right]^{-1}$$

  - Mean free path λ as a function of radiation type, energy, and material
  - For creation of secondary packets, double-differential cross-section data may be required for some isotopes and reactions channels

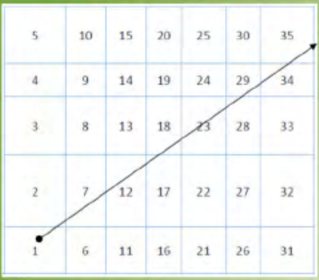
Larry Cox, ljcox@lanl.gov, (505) 665-7344
Ryan Marcus, rmarcus@lanl.gov, (505) 500-4365

LA-UR-11-06271

Monte Carlo mini-App — Main Computational Steps

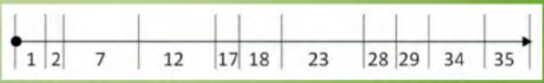
The computation is broken into these separable steps each with its own data needs and memory footprints:

1. **Source:** generate random packets
   (position, direction, energy, type, weight...)



2. **Ray Trace:** trace packets through the geometry
   - To determine geometry elements intersected

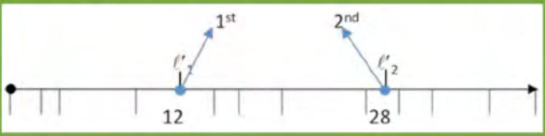
3. **Path Length:** calculate path lengths in each intersected element
   - Path length is the fundamental quantity to be scored



4. **Attenuation:** reduce the weight along the track

$$w_{i+1} = w_i e^{-ds/\lambda}$$

5. **Kinematics:** spawn secondary packets



6. **Accumulate:** score results of interest
   (flux, image, secondaries...)

Larry Cox, ljcox@lanl.gov, (505) 665-7344
Ryan Marcus, rmarcus@lanl.gov, (505) 500-4365

Monte Carlo mini-App — It's a wrap…

Next Steps

There is lots left to do. Some of the plans for the coming year include:

- Definition of simple (non-physical) materials and cross sections
  - For distribution with the mini-App
  - For use with MCNP6 for comparisons
- Extension of the Exa Python framework to additional parallelism paradigms and operations, as needed
- Source and image plane methods for imaging applications (radiography, etc.)
- Iterative sources for applications such as criticality safety
- And lots more…

If you are interested in getting involved or in trying out these MC mini-App concepts, please get in touch.

References:

[1] MCNP website: http://mcnp.lanl.gov

[2] L.J. Cox, R. Marcus, *Developing a Monte Carlo mini-App for Exascale Co-Design*, LA-UR-2011-06086

[3] R. Marcus, L.J. Cox, *Python Framework for Co-Design Applications in Exascale R&D*, LA-UR-2011-06085

Larry Cox, ljcox@lanl.gov, (505) 665-7344
Ryan Marcus, rmarcus@lanl.gov, (505) 500-4365