An Update of ARCHER, a Monte Carlo Radiation Transport Software Testbed for Emerging Hardware Such as GPUs

X. George Xu*[1], Tianyu Liu[1], Lin Su[1], Xining Du[1], Matt Riblett[1], Wei Ji[1], Forrest B. Brown[2]

[1]*Nuclear Engineering Program, Rensselaer Polytechnic Institute, Troy, NY, 12180; email: xug2@rpi.edu*
[2]*Monte Carlo Codes Group, Los Alamos National Laboratory, Los Alamos, NM, 87545*

INTRODUCTION

Heterogeneous computing systems involving the graphics processing unit (GPU) and other accelerators such as the coprocessor are playing an increasingly important role in scientific computing. However, none of the existing production Monte Carlo (MC) radiation transport codes were designed to take advantage of such heterogeneous computer architectures.

In this paper, we describe the development of a software package, called **ARCHER - A**ccelerated **R**adiation-transport **C**omputations in **H**eterogeneous **E**nvi**R**onments, which is designed as a software testbed for research on how to accelerate Monte Carlo calculations using the Nvidia GPU and the Intel Xeon Phi coprocessor.

DESCRIPTION OF THE ACTUAL WORK

ARCHER has been used to investigate a variety of topics: 1) CT imaging dose, 2) electron dose, 3) criticality calculation, 4) vectorized MC algorithms, and 5) comparison of the Nvidia GPU and the Intel Xeon Phi in terms of performance as Monte Carlo accelerator.

In each study, we first develop a serial code $\text{ARCHER}_{\text{CPU}}$ in C as a basis for performance comparison. Then we modify the code to $\text{ARCHER}_{\text{GPU}}$ and $\text{ARCHER}_{\text{MIC}}$ which are specific to the Nvidia GPU and the Intel MIC architecture, respectively. The GPU code is written in NVIDIA's CUDA C and the MIC code in Intel's heterogeneous offload programming model. When porting the code from CPU to GPU, two kernels were implemented: one for the random number generator (RNG) and the other for the particle transport. We employed the Xorshift algorithm to generate pseudo-random numbers on the fly. Each CUDA thread kept an independent local RNG state variable for generating a sequence of random numbers, such that sequences from different threads are statistically uncorrelated. The transport kernel for a certain type of particle is executed in parallel. The code porting from CPU to MIC was realized by applying the offload compiler directives to the parallelizable part of the CPU code. In practice, the way to perform memory management is slightly different from the GPU runtime API in that, 1) data movement and persistence is realized by offload directives, 2) except for the phantom array, each hyper-thread acquires an individual copy of cross-section and indexing data.

ARCHER contains voxelized human patients including the RPI-Adult Male, Female and Obese Phantoms. The GPU-based code is capable of performing accurate and fast radiation dose calculation for a voxelized phantom with primary particles being either photons or electrons. Compton scattering, Rayleigh scattering, photoelectric and pair production are considered for photons with energy range from 1 keV to 20 MeV. We used the woodcock tracking algorithm for photon transport in heterogeneous media. Since our previously published work [1], our photon transport code has been considerably expanded for accommodating more physics. Now there are two sets of physics models available in ARCHER for photon transport: 1) Detailed physics, where we took into account the effects of atomic form factors, and calculated the material attenuation coefficients on the fly from microscopic cross sections. 2) Simply physics, where we ignored the atomic form factors and used a pre-tabulated cross section for each material. The energy-dependent cross-section was linearly interpolated.

Another new feature we added in ARCHER is a module of handling electron transport in voxelized geometry. Previously in our code electrons were not transported and their energies were simply locally deposited. The current code provides the option of enabling electron transport in heterogeneous media, a process which could be important for dealing with electrons with high energies. Electron transport in ARCHER is based on the class-II condensed history method and continuously slowing down approximation (CSDA). Moller scattering and Bremsstrahlung are modeled for interaction with energy loss greater than the critical value.

To investigate the applicability of vectorized MC on GPUs, we developed the GPU-based MC code for a neutron eigenvalue problem [2] using the event-based vectorized algorithm, which helps to eliminate the branch divergence issue that lies in the traditional history-based algorithm on GPUs. Details of the vectorized MC algorithm can be found in Ref [3]. For simplicity we used the one-speed approximation and considered three physical processes including elastic scattering, fission and capture. Generalization to multi-group or continuous energy involves additional group bin search processes and may suffer more from the thread divergence and memory latency problems on GPUs. The development of such a GPU code is subject to future studies.

RESULTS

Several tests have been performed to benchmark the efficiency of ARCHER code for various medical physics and nuclear engineering applications.

In one case, we consider a whole body helical CT scan over a 142-kg RPI Adult Male obese phantom, using a scan protocol that includes 120kVp, 20mm beam collimation and a pitch of 1. A validated GE LightSpeed 16 multiple detector CT scanner model was utilized for the MC simulation. Because the CSDA range of secondary electrons inside the human body at the considered energy scope is generally one order of magnitude smaller than the dimension of a voxel, their transport process was safely ignored.

The GPU codes and MIC codes were tested on a NVIDIA Tesla M2090 card and an Intel Xeon Phi coprocessor, respectively. The CPU codes were tested on an Intel Xeon X5650 2.66GHz CPU with 16 GB RAM.

Table I. Execution time of various codes for the CT scan simulation. Here "Diff" represents the percentage difference from MCNPX results defined by Diff = |results - results$_{MCNP}$| / results$_{MCNP}$×100%.

CODE	SIMPLE PHYSICS		DETAILED PHYSICS	
	Diff [%]	Time(min)	Diff[%]	Time(min)
ARCHER$_{CPU}$(6 cores)	4.45	24.2	0.42	138.6
ARCHER $_{GPU}$	1.93	7.4	0.38	56.0
ARCHER$_{MIC}$	4.45	13.8	0.32	41.6

In Table I, we show the simulation time for 10^9 photons using various codes. For a relatively fair comparison, we show the CPU results of the OpenMP run on 6 cores of the Intel Xeon X5650 CPU. The ARCHER$_{GPU}$ and ARCHER$_{MIC}$ with simplified physics are 3.3 and 1.8 times, respectively, faster than the OpenMP ARCHER$_{CPU}$ code. These numbers vary for codes with detailed physics. In both cases, one can see significant speed up factors resulted from using accelerators like GPUs and Intel Phi coprocessors. For the same test, MCNPX running on CPU in sequential mode takes 8689 minutes, which is ~1200 times slower than the ARCHER$_{GPU}$ code with simplified physics. Note that this is partially due to the fact that MCNPX employs more sophisticated physics models than ARCHER does, such as Doppler broadening and X-ray fluorescence.

In another test, we studied the electron dosimetry for a water-aluminum-water slab geometry. The phantom is a 30 × 30 × 30 cm^3 cube with voxel size of 0.5 × 0.5 × 0.2 cm^3. Along z axis, there are three levels: 2cm water, 1 cm aluminum and 37 cm water. The radiation source used in simulation is 20 MeV monoenergetic parallel electron beam incident perpendicular to the x-y plane with a field size of 3.0 × 3.0 cm^2. A total of 6×10^6 histories were simulated. The depth dose of the central axis and lateral dose of different depth (z = 1 cm, z = 4.8 cm, z = 7 cm) were tallied.

Table II: Simulation time for 6×10^6 20 MeV electrons incident on a water-aluminum-water phantom.

CODE	TIME [SEC]
MCNPX	9213
EGSnrc	1645
ARCHER$_{CPU}$	186
ARCHER$_{GPU}$	7.33

Simulation time for various codes was listed in Table II. The ARCHER$_{GPU}$ code is ~25 times faster than the sequential ARCHER$_{CPU}$ code, and this demonstrates the great efficiency of using GPUs to accelerate MC simulations. Again, part of the reason for the long simulation time of MCNPX and EGSnrc in sequential mode is that they are using more complex physics models than ARCHER does.

In the third test, we compared the performance of various codes for a simple neutron eigenvalue problem. The geometry considered is a heterogeneous 1D system consisting 10 fuel slabs and 11 moderator slabs. We used the parameter set $\Sigma_F = 0.034$ cm^{-1}, $\Sigma_A = 0.08$ cm^{-1}, $\Sigma_T = 0.1$ cm^{-1}, $\nu = 2.5$, $\Delta_x = 3.8$ cm for the fuel, and $\Sigma_A = 0.0001$ cm^{-1}, $\Sigma_T = 0.1$ cm^{-1}, $\Delta_x = 30.0$ cm for the moderator. We found that the vectorized code, although significantly enhancing the warp execution efficiency, is 10 times slower than the conventional history-based code on GPUs. The profiling results suggest that this is mainly due to high cost of global memory transactions.

CONCLUSION

Applications of ARCHER to nuclear engineering and medical physics problems have shown exciting results. Using hardware accelerators like GPUs and the Intel Phi coprocessors, we have achieved significant speed up factors over our CPU code and other production MC codes. On-going efforts aim to implement more complex physics models and further optimization techniques to improve both accuracy and performance of the code.

ACKNOWLEDGEMENT

This research was partially supported by the grant from NIBIB (R01EB015478).

REFERENCES

1. T. LIU, A. DING, X.G. XU, "GPU-Based Monte Carlo Methods for Accelerating Radiographic and CT Imaging Dose Calculations: Feasibility and Scalability," *Med. Phys.*, 39(6):3876, (2012).
2. T. LIU, A. DING, W. JI, X. G. XU, C. D. CAROTHERS, F. B. BROWN, "A Monte Carlo Neutron Transport Code for Eigenvalue Calculations on a Dual-GPU System and CUDA Environment", *Physor 2012 Advances in Reactor Physics*, Knoxville, TN, USA, April 15-20, 2012.
3. X. Du, T. Liu, W. Ji, X. G. Xu, "Evaluation Of Vectorized Monte Carlo Algorithms on GPUs for a Neutron Eigenvalue Problem", International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013), Sun Valley, Idaho, USA, May 5-9, 2013.