

Photon Propagation with GPUs in IceCube

Dmitry Chirkin¹ for the IceCube collaboration*

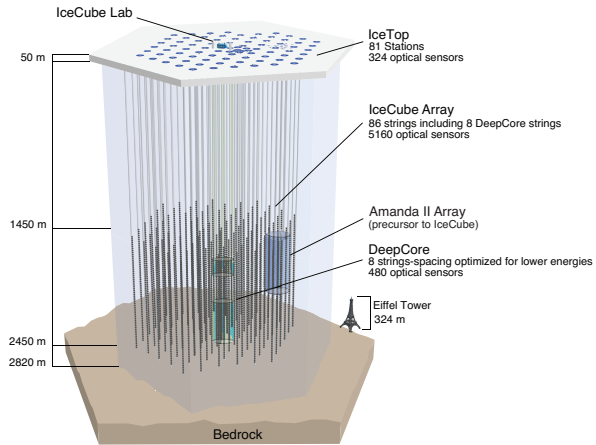
¹UW-Madison, Madison, WI, U.S.A.

DOI: <http://dx.doi.org/10.3204/DESY-PROC-2014-05/40>

A typical muon event in IceCube creates in excess of 10^7 Cherenkov photons in the sensitive wavelength range, presenting a considerable computational challenge when more than a billion of such muons need to be simulated to represent just a few days of the detector data. However, we note that describing propagation of a large number of photons in a transparent medium is a computational problem of a highly parallel nature. All of the simulated photons go through the same stages: they are emitted, they may scatter a few times, and they get absorbed. These steps, when performed in parallel on a large number of photons, can be done very efficiently on a GPU. The IceCube collaboration uses parallelized code that runs on both GPUs and CPUs to simulate photon propagation in a variety of settings, with significant gains in precision and, in many cases, speed of the simulation compared to the table lookup-based code. The same code is also used for the detector medium calibration and as a part of an event reconstruction tool.

1 Introduction

IceCube (shown on the right, see [1]) is a cubic-kilometer-scale high-energy neutrino observatory built at the geographic South Pole. IceCube uses the 2.8 km thick glacial ice sheet as a medium for producing Cherenkov light emitted by charged particles created when neutrinos interact in the ice or nearby rock. Neutrino interactions can create high-energy muons, electrons or tau leptons, which must be distinguished from a background of downgoing atmospheric muons based on the pattern of emitted Cherenkov light. The photons that reach the detector have wavelengths in the visible range (300 to 550 nm), are emitted at the Cherenkov angle of about 41° , and are subject to both scattering (with scattering length varying between values of 5 to 90 m within the detector volume) and absorption (20-280 m) [2]. Since scattering and absorption lengths



*<http://icecube.wisc.edu>

are comparable to each other, and to the typical distances at which the events are observed (20-150 m), existing analytical approximations (single scattering or, on the other end, diffusive) cannot be used, and the photons must be propagated individually in the simulation. Results of such propagation can optionally be tabulated, but the parameter space that must be explored in such an approach is large (in 8 dimensions or more), necessitating certain simplifications. Additionally, the tables are slow to generate, quite large, and may still suffer from certain issues (e.g., related to binning or interpolation strategy).

2 Photon Propagation Software

The photon propagation code (PPC) [3] was initially written to study the feasibility of direct photon propagation for simulation of events in IceCube. The simple nature of photon propagation physics allowed us to focus on the code optimization, to make sure the simulation ran as fast as possible. The simulation was written in C++, then re-written entirely in Assembly for the 32-bit i686 architecture with SSE¹ vector optimizations. The Assembly version of the program used the SSE instructions for updating photon direction when scattered and for locating the optical sensors near the photon segment, while the calculation of the scattering angle was performed by keeping all intermediate variables inside the registers of the float point unit (FPU) stack. The calculation speed improved by 25-37% compared to the C++ version.

Shortly thereafter we were able to demonstrate that significant acceleration of the photon propagation by factors of 150 or more (compared to running on a single CPU core) is possible by using graphics processing units (GPUs). We confirmed this with a version of software that employs the Nvidia GPUs (graphics processing units) via the CUDA² programming interface, and a second version that uses OpenCL³, supporting both Nvidia and AMD GPUs, and also multi-CPU environments. The IceCube collaboration also uses a software module called CLSIM⁴ that was independently written for OpenCL, and verified to produce results statistically indistinguishable from those produced with PPC.

The reason for the substantial acceleration of our simulation on GPUs is the highly parallel nature of the simulation of the photon propagation. All of the simulated photons go through the same steps before getting absorbed or hitting a sensor (see Figure 1): photon propagation between the scattering points, calculation of the scattering angle and new direction, and evaluation of whether the current photon segment intersects with any of the optical sensors of the detector array. The GPUs are designed to perform the same computational operation in parallel across multiple threads. Each thread works on its own photon for as long as the photon exists. When the photon is absorbed or hits the detector the thread receives the new photon from a pool of photons for as long as that pool is not empty. Although a single thread runs slower than a typical modern computer CPU core, running thousands of them in parallel results in the much faster processing of photons from the same pool on the GPU.

¹Streaming SIMD Extensions, an extension to the CPU instruction set that operates on vector data.

²Compute Unified Device Architecture, a programming model by Nvidia that provides access to the computing features of their GPU cards.

³Open Computing Language, a framework for accessing features of devices working on multiple data elements in parallel. This includes devices like CPUs, GPUs, and accelerator cards from multiple manufacturers.

⁴OpenCL-based photon-tracking simulation modeling scattering and absorption of light in ice or water.

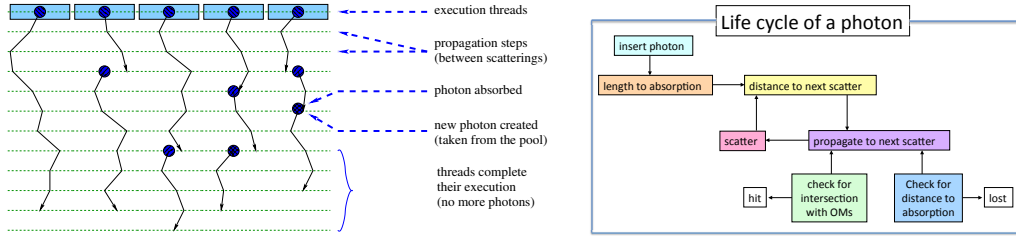


Figure 1: Parallel nature of the photon propagation simulation: tracking of photons entails the computationally identical steps: propagation to the next scatter, calculation of the new direction after scatter, and evaluation of intersection points of the photon track segment with the detector array. These same steps are computed simultaneously for thousands of photons.

3 Simulation with photon propagation code

The direct photon simulation software is typically used in the two scenarios shown in Figure 2. In the first scenario the in-situ light sources of the detector are simulated for calibrating the detector and the properties of the surrounding ice. It is possible to very quickly re-simulate the detector response to a variety of ice scattering and absorption coefficients which are finely tabulated in depth bins. This allows for these coefficients to be fit directly, by finding the combination that is a best simultaneous fit to all of the in-situ light source calibration data [2]. For the 10 meter depth bins, 200 coefficients are fitted (with scattering and absorption defined in 100 layers spanning 1 km of depth of the detector), with nearly a million possible ice parameter configurations tested in less than a week on a single GPU-enabled computer. This method is intractable with a table-based simulation, as each new parameter set would require generation of the new set of parametrization tables, each generation taking on the order of a week of computing time on a ~ 100 -CPU cluster.

In the second scenario the Cherenkov photons created by the passing muons and cascades are simulated as part of the larger simulation of the detector response to atmospheric and other fluxes of muons and neutrinos. The simulation is able to account for some effects that are difficult to implement with the table-based simulation, because their simulation would lead to additional degrees of freedom, thus increasing the size of the parametrization effort and tables many-fold. One of these is the tilt of the ice layers, *i.e.*, dependence of the ice parameters not only on the depth, but also on the xy surface coordinates [2]. Another effect, recently discovered, is ice anisotropy [4], which manifests itself as a roughly 13% variation in scattering length depending on the orientation.

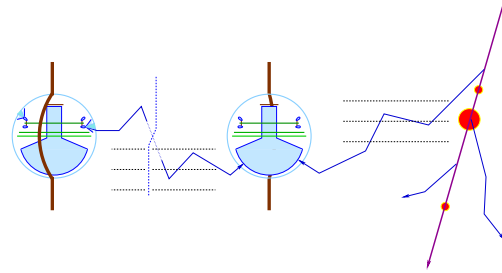


Figure 2: Typical simulation scenarios: photons emitted by the detector are tracked as part of the calibration procedure (left). Cherenkov photons emitted by a passing muon and cascades along its track are tracked to simulate the typical IceCube events (right).

4 IceCube GPUs

The IceCube collaboration operates or otherwise has access to multiple GPU clusters. Recent work on fitting the ice was done at a cluster of 32 Nvidia 690 (2 GPUs each) and 32 AMD 7970 cards (for 96 GPUs altogether). The fitting works by simulating several slightly different ice configurations, waiting for the results, and finding the best configuration. Then creating several slightly modified configurations (based on the best configuration calculated in the previous step), and so on.

We have studied the relative performance of different GPU cards currently in use within our collaboration and summarize our findings in Table 1.

To optimize the use of the GPU-enabled computers further we employ the DAG⁵ tools. This involves separating simulation segments into tasks, and assigning these tasks to DAG nodes. DAG assigns separate tasks to different computer nodes; execution of photon propagation simulation is performed on dedicated GPU nodes.

Card	Performance	TDP
AMD 7970	1.77	250
AMD 7950	1.57	200
Nvidia 690	2	300
Nvidia 680	1	195
Nvidia 660	0.62	140
Nvidia 650 ti	0.43	110
Nvidia m2070	0.66	225

Table 1: Relative performance of the tested GPU cards. TDP is the manufacturer's specified maximum card heat generation in Watts.

5 Concluding remarks

We report on a photon propagation tool that replaces the older table-based approach in certain situations, while achieving more precision, better description of the physics of the process and shorter run time. This tool is capable of running on both CPU cores and GPU hardware, achieving significant speed up (factors in excess of ~ 100) on the latter.

References

- [1] M. G. Aartsen and others (IceCube collaboration). Evidence for high-energy extraterrestrial neutrinos at the icecube detector. *Science*, 342:1242856, 2013.
- [2] M. G. Aartsen et al. (IceCube collaboration). Measurement of south pole ice transparency with the icecube led calibration system. *Nucl. Inst. Meth. A*, 711:73–89, 2013.
- [3] Dmitry Chirkin for the IceCube Collaboration. Photon tracking with gpus in icecube. *Nucl. Inst. Meth. A*, 725:141–143, 2013.
- [4] Dmitry Chirkin for the IceCube Collaboration. Evidence of optical anisotropy of the south pole ice. *proceedings to 33th ICRC, Rio de Janeiro*, 2013, arXiv:1309.7010.

⁵Directed Acyclical Graph, used to represent dependencies in a software chain