



Contribution ID: 40

Type: **not specified**

A study on implementing a multithreaded version of the SIRENE detector simulation software for high energy neutrinos

Tuesday, 15 September 2015 17:10 (20 minutes)

Aim of this work is the development of a multithreaded version of the SIRENE detector simulation software for high energy neutrinos. This approach allows utilization of multiple CPU cores and GPUs, leading to a potentially significant decrease in the required execution time compared to the sequential code. We are making use of MPI, OpenMP frameworks for the production of multithreaded code running on the CPU and CUDA framework to leverage the processing power implicating the GPU. SIRENE implements different geometries for a neutrino detector and different configurations and characteristics of photo-multiplier tubes (PMTs) inside the optical modules of the detector through a library of C++ classes. This could be considered a massive statistical analysis of photo-electrons. Each event consists of a number of particles (tracks) in the detectable area, each track represents the different energy, direction and time of arrival of each particle. Energy loss is calculated in steps. For each step the probability of reaching an optical module is calculated and for each of these modules the number of photo-electrons that give a hit is calculated for each of the PMTs inside.

Accordingly, MPI could be used for the parallelization of the events since they are independent of each other and there is no data exchange between them whatsoever. This permits computations to be trivially spread over several processing nodes. OpenMP could be used to parallelize the tracks, which, in the original sequential code, is the largest external loop containing computations for each particle. It could be implemented by enlisting one thread per track capable of creating sub-threads as needed, according to the inner loops and the available system resources. The most critical part of the sequential code is the loop referring to the energy loss and involving the final calculation which needs to be transformed in order to allow parallel execution of the loop. In-between, certain parts of SIRENE could be executed using CUDA. The coordinate system must be defined such that the track direction is pointed along the z -axis and the position of the module is located in the x - y plane. The rotation of the coordinate system is expressed as a 3×3 matrix and is used on every module, making this part ideal for acceleration via GPU. Furthermore, Poisson distribution is employed to calculate the arrival times of photo-electrons on a module and polynomial interpolation is repeatedly used during the computation of the number of photo-electrons that hit a PMT. In both cases, there is a choice of either applying straight parallelization to the existing sequential algorithm, or, if this approach does not offer acceptable results in terms of suitability and speed, look into alternate parallel versions of those algorithms. It is also possible to take advantage of fast hardware implementations of arithmetic functions such as hardware linear interpolation on the GPU.

Primary author: GIANNAKOPOULOS, Petros (University Of Athens)

Co-authors: TZAMARIOUDAKI, Aikaterini (Nat. Cent. for Sci. Res. Demokritos (GR)); FILIPPIDIS, Christos (Nat. Cent. for Sci. Res. Demokritos (GR)); MARKOU, Christos (Nat. Cent. for Sci. Res. Demokritos (GR)); VOULARINOS, Georgios (University of Athens); DIPLAS, Ioannis (University of Athens); BALASI, Konstantia (Nat. Cent. for Sci. Res. Demokritos (GR)); GKOUUMAS, Michail (University of Athens); Prof. COTRONIS, Yiannis (University of Athens)

Presenter: FILIPPIDIS, Christos (Nat. Cent. for Sci. Res. Demokritos (GR))

Session Classification: Parallel Session G