NEC'2019
XXVII International Symposium on Nuclear Electronics & Computing
Montenegro, Budva, Becici, 30 September - 4 October 2019

Contribution ID: **249** Type: **Sectional**

Accelerating the particle-in-cell method of plasma and particle beam simulation using CUDA tools

Thursday 3 October 2019 12:00 (15 minutes)

For simulating the dynamics of charged particles in electric and magnetic fields a particle-in-cell (PIC) method is often used. In it the position and velocity of each particle or superparticle is tracked, while the charge density and current density necessary to simulate particle interactions are computed on a stationary mesh. Several approaches are availible to integrating the particle equations of motion (particle mover) and calculating the electric and magnetic fields (field solver).
The Ef and Ef_python applications aim to use particle-in-cell to simulate plasma and particle beams in an electron beam ion source. The main goal, unlike many other PIC applications, is not to simulate free plasma, but ion sources. As the particle mover the well-established leapfrog second-order explicit method is used, also known as the Boris scheme. The electric field created by charged particles and conducting regions is calculated using the finite-difference method of solving the Poisson equation on a rectangular regular grid.
Each PIC simulation time step consists of the following operations: advance particle positions and momenta, generate new particles, calculate charge density, compute electric potential, calculate electric field. The simulation performance for each step depends on the number of particles, as well as the number of spatial mesh cells, with different operations dominating the performance considerations. The field solver usually represents a major portion of computational difficluty.
This report describes the efforts to profile and accelereate the the ef_python application using the cupy library. With minimal changes to the program, cupy allowed to perform most of the simulation operations on the GPU through CUDA. In addition, to accelerate the field solver algebraic multigrid methods were utilized, provided by the PyAMG library on the CPU and the AMGX library on the GPU. Major speed-up was achieved especially with fine spatial grids and a powerful GPU in the HybriLIT cluster.

Primary author: Mr KADOCHNIKOV, Ivan (JINR)

Presenter: Mr KADOCHNIKOV, Ivan (JINR)

Session Classification: Machine Learning Algorithms and Big Data Analytics

Track Classification: Computations with Hybrid Systems (CPU, GPU, coprocessors)