Dynamic load-balancing and GPU computing with the particle-in-cell code PSC

K. Germaschewski[1], H. Ruhl[2], Will Fox[1], and A. Bhattacharjee[1]

[1] Department of Physics, University of New Hampshire, Durham, NH, USA

[2] Department of Physics, Ludwig-Maximilians-Universität, München, Germany

We have developed a new version of the Particle Simulation Code (PSC), originally written by H. Ruhl. The new code is designed with state-of-the-art and future massively parallel high-performance computers in mind, and has extensible support for various physics modules, e.g., modeling collisions and QED effects. At its core, the code uses the explicit particle-in-cell method to solve the relativistic Vlasov-Maxwell equations in 3D and in reduced dimensions.

We will present scaling results on Cray and IBM Bluegene supercomputers up to 100k cores. Load imbalance that develops over the course of a simulation is a serious problem for particle-in-cell codes as particles move across the domain and aggregate in certain areas, locally increasing the load. Adapting the decomposed domain by moving the processor boundaries along coordinate axes is only moderately effective in rebalancing the load. We have developed a novel dynamic load balancing method based on space-filling curves that reduces the maximum imbalance of a sample production run from a factor of larger than 2 to just a few percent.

Graphics Processing Units (GPUs) have shown large promise in achieving substantially enhanced performance over conventional processors, but it is hard to find particle-in-cell algorithms that efficiently exploit the fine-grained parallelism provided through nvidia's CUDA programming model. We will present different algorithms and evaluate the performance achieved. In particular, we will focus on different approaches of depositing the current onto the grid, which is the most difficult part of the method to implement in a highly threaded manner due to concurrent read-write-update operations.