Adaptable Particle-in-Cell Algorithms for Graphical Processing Units

Viktor K. Decyk\textsuperscript{1,2} and Tajendra V. Singh\textsuperscript{2}

\textsuperscript{1}Department of Physics and Astronomy, University of California, Los Angeles, Los Angeles, CA 90230, USA
\textsuperscript{2}Institute for Digital Research and Education, University of California, Los Angeles, Los Angeles, CA 90230, USA

decyk\_at\_physics.ucla.edu

Emerging computer architectures consist of an increasing number of shared memory computing cores in a chip, often with vector (SIMD) co-processors. Future exascale high performance systems will consist of a hierarchy of such nodes, which will require different algorithms at different levels. Since no one knows exactly how the future will evolve, we have begun development of an adaptable Particle-in-Cell (PIC) code, whose parameters can match different hardware configurations \cite{1}. The data structures reflect three levels of parallelism, contiguous vectors and non-contiguous blocks of vectors, which can share memory, and groups of blocks which do not. Particles are kept ordered at each time step, and the size of a sorting cell is an adjustable parameter. We have implemented a relativistic 2-1/2D electromagnetic skeleton code whose inner loop (containing 10 subroutines) runs entirely on the NVIDIA Tesla C1060. The performance obtained varies from 2.8 to 3.7 nsec/particle/time step, depending on the plasma temperature. This is a speedup of about 33-44 compared to a single 2.66 GHz Intel i7 (Nehalem) processor, which has a performance of 123 nsec/particle/time step.