Japan Geoscience Union Meeting 2010
(May 23-28 2010 at Makuhari, Chiba, Japan)

PEM030-14 Room: 303 Time: May 26 11:05-11:22

OhHelp Load Balancer for PIC Simulation and Its Library

Hiroshi Nakashima[1*]

[1]Kyoto University

We have developed a new method for Particle-in-Cell (PIC) simulations which aims at achieving both good load balancing and scalability so as to be efficiently executed on distributed memory systems. This method, named OhHelp, has unique problem decomposition and load balancing mechanisms which are outlined as follows.

+ The space domain is equally partitioned to assign each subdomain to each computation node as its primary subdomain. If the number of particles in each subdomain is well-balanced, i.e., less than the average plus a certain tolerance, each node is only responsible its primary subdomain and the particles residing in it.

+ If one or more subdomains have too many particles, every but one node is responsible of another subdomain which has particles more than average as its secondary subdomain.

+ A part of particles in the secondary subdomain of a helper node are assigned to the node from its helpand so that no nodes have too many particles. The OhHelp load balancer dynamically examines the sustainability of the helpand-helper configuration and determines the number of primary and secondary particles of each node to keep good balancing.

Since a node has to have at most two subdomains, OhHelp is scalable with respect to the domain size. As for the number of particles, OhHelp keeps its excess over the per-node average less than the tolerance by dynamically rearranging the secondary subdomain assignment and thus also achieves good scalability.

We have applied the OhHelp method to two types of PIC simulators, a full-particle simulator[1] and a particle-fluid hybrid simulator[2]. Both implementations achieve good scalability exhibiting almost linear speed-up with 256 processes, 166-190 fold in the full-particle one and 242-260 fold in the hybrid simulation, for a large system of 256x256x128 grids and more than two billion particles residing in it. A notable fact is that the 166-fold and 242-fold speed-ups are for an extremely non-uniform particle distribution in which all the particles are densely packed in a small space of 1/256 of the whole space domain and travel across it causing not only severe imbalance of the particle population among subdomains but also its fluctuation. These good speed-ups even in the extreme case, or the small performance degradation from the 190-fold and 260-fold in the happiest case of perfectly uniform particle distribution, proves that OhHelp should be always efficient with any types of simulations.

Another important feature of OhHelp is that it is not for our own simulators but is intended to be applied to any PIC simulators. Our OhHelp source code package provides PIC people with a layered set of library functions with Fortran and C interfaces. Its level-1 functions, for those who have already developed their own domain-decomposed simulators but suffer from inefficiency due to load imbalance, will show them how many particles have to be transferred between every pair of computation nodes. If it is tiresome for you to perform sending and receiving operations for particle transfer, level-2 functions will do it for you taking care of particle injections and removals. Furthermore, the OhHelp library gives you great help even if your simulator is not parallelized at all, since its rich set of level-3 functions performs boundary data exchange between adjacent subdomains, reduction operations to have charge and/or current density contributed to each grid point by particles among nodes, and so on. The OhHelp source code package together with an extensive user's manual is available at our web site http://www.para.media.kyoto-u.ac.jp/ohhelp/.

[1] H. Nakashima, et al.: OhHelp: A Scalable Domain-Decomposing Dynamic Load Balancing for Particle-in-Cell Simulations, In ICS'09, 2009.

[2] J. Akiyama, et al.: A Parallelization of Particle-Fluid Hybrid Plasma Simulation with the OhHelp Load Balancer, In IPSJ Tech. Rep. 2010-HPC-124, 2010.

Keywords: PIC simulation, parallel simulation, load balancing