The OpenFPM framework for SPH and particle-mesh simulations on heterogeneous parallel computers

Pietro Incardona[1,2], Ivo F. Sbalzarini[1,2]

[1] Chair of Scientific Computing for Systems Biology, Faculty of Computer Science, TU Dresden, 01069 Dresden, Germany

[2] Center for Systems Biology Dresden, Max Planck Institute of Molecular Cell Biology and Genetics, Pfotenhauerstr. 108, 01307 Dresden, Germany

Scalable Smoothed Particle Hydrodynamics (SPH) and particle-mesh simulations gain importance as fundamental research tools, and they are at the base of many discoveries alongside theory and experiment. Meanwhile, the performance of computing hardware continues to grow, mainly by increasing parallelism, enabling simulations of ever more complex problems. However, the development of scalable codes able to run on heterogeneous, distributed hardware systems increasingly becomes the bottleneck. This bottleneck can be alleviated by intermediate software layers that provide higher-level abstractions closer to the problem domain of particle methods, like SPH, hence allowing the computational scientist to focus on the simulation method rather than the intricacies of the underlying hardware and to require less time for code implementation.

13 years ago, the PPM library introduced an abstraction layer for numerical simulations using particles and/or meshes on distributed systems of CPUs. The PPM library has since been used for particles-only and hybrid particle-mesh simulations of both discrete and continuous models, as well as non-simulation applications, such as image segmentation. Unfortunately, however, PPM lacks portability to other hardware platforms and does not support space dimensions larger than 3, nor complex particle properties. We therefore present OpenFPM, an open and scalable C++ framework that provides similar top-level abstractions as PPM, but removes the aforementioned limitations of PPM.

OpenFPM is complemented with frequently used numerical routines, as well as interfaces to third-party libraries. We present the architecture and design of OpenFPM, detail the underlying abstractions, and benchmark the framework in applications including SPH. We show how OpenFPM transparently supports accelerator hardware, such as GPUs, and how it can be used to implement fully scalable and portable SPH codes in just a few days.