Distributed Memory Fluid-Solid Interaction Simulations via Chrono::HPC

We present a distributed memory framework for large-scale fluid-solid interaction simulations called Chrono::HPC. In this simulation environment the method of Smoothed Particle Hydrodynamics (SPH) is employed to simulate the dynamics of fluid interacting with 3D rigid bodies. The implementation relies on Charm++, an asynchronous message passing parallel programming framework able to run in a distributed memory/cluster setup. Charm++ is an object-based programming paradigm whose work and data units are organized as C++ objects called *chares*. We employ a hybrid decomposition strategy that targets the parallelization of both the physical domain and force computation, where the physical domain is divided into cells referred to herein as cell chare objects. Force computations are performed using compute chare objects that are assigned to pairs of interacting cells as well as the cell itself. The sets of cell and compute chares are organized into their respective chare arrays. A cell chare array is a dense 3D array whose indices (x, y, z) have a one-to-one mapping to the spatial decomposition of the problem. Compute chares are organized into a sparse 6D chare array. Each component of this array, indexed as (x1, y1, z1, x2, y2, z2), handles the interaction of cell pairs with indices (x1, y1, z1) and (x2, y2, z2). This hybrid decomposition scheme allows the implementation to follow three of the core components of the Charm++ parallel programming paradigm: object-oriented (cell and compute chares), over-decomposition, and asynchronous message-driven execution. The over-decomposition component of Chrono::HPC comes into play when minimizing the compute idle time. The number of cell and compute chares is selected to be greater than the number of available computational cores. This allows the Charm++ Runtime System (RTS) to assign multiple chares to each core. Charm++ RTS chooses to alternate between chares when they are waiting for data, which hides memory latency by overlapping computation and distributed memory access. This over-decomposition decision is motivated by Charm++'s asynchronous message-driven nature. An additional chare array handles the rigid body dynamics by (a) processing the reduction of distributed fluid forces on each body, and (b) advancing its state in time through numerical integration. The Chrono::HPC framework is demonstrated via a simulation of a dam-break problem containing buoyant rigid bodies and scaled to systems with tens of millions of degrees of freedom. The results obtained indicate linear strong scaling of the simulation engine, which demonstrates if ability to leverage supercomputers to simulate FSI problems with dimensions larger than those tested herein.