Offloading computational tasks of hybrid MPI + OpenMP/OmpSs-2 applications to GPUs

Table of contents:

Quick Overview

NBody Benchmark

Users can clone or download this examples from the ?https://pm.bsc.es/gitlab/DEEP-EST/apps/NBody repository and transfer it to a DEEP working directory.

Description

An NBody simulation numerically approximates the evolution of a system of bodies in which each body continuously interacts with every other body. A familiar example is an astrophysical simulation in which each body represents a galaxy or an individual star, and the bodies attract each other through the gravitational force.

N-body simulation arises in many other computational science problems as well. For example, protein folding is studied using N-body simulation to calculate electrostatic and *Van der Waals* forces. Turbulent fluid flow simulation and global illumination computation in computer graphics are other examples of problems that use NBody simulation.

Requirements

The requirements of this application are shown in the following lists. The main requirements are:

- **GNU Compiler Collection**.
- **OmpSs-2**: OmpSs-2 is the second generation of the **OmpSs** programming model. It is a task-based programming model originated from the ideas of the OpenMP and StarSs programming models. The specification and user-guide are available at ?https://pm.bsc.es/ompss-2-docs/spec/ and ?https://pm.bsc.es/ompss-2-docs/user-guide/, respectively. OmpSs-2 requires both **Mercurium** and **Nanos6** tools. Mercurium is a source-to-source compiler which provides the necessary support for transforming the high-level directives into a parallelized version of the application. The Nanos6 runtime system library provides the services to manage all the parallelism in the application (e.g., task creation, synchronization, scheduling, etc). Downloads at ?https://github.com/bsc-pm.
- **Clang + LLVM OpenMP** (derived):
- **MPI**: This application requires an MPI library supporting the multi-threading level of thread support.

In addition, there are some optional tools which enable the building of other application versions:

- **CUDA** and NVIDIA **Unified Memory** devices: This application has CUDA variants in which some of the N-body kernels are executed on the available GPU devices.
- **Task-Aware MPI (TAMPI)**: The Task-Aware MPI library provides the interoperability mechanism for MPI and OpenMP/OmpSs-2. Downloads and more information at ?https://github.com/bsc-pm/tampi.