



Contribution ID: 49

Type: **Talk (15min + 5min)**

SPHinXsys Parallelization with SYCL: Smoothed Particle Hydrodynamics on Heterogeneous Systems

Wednesday, March 6, 2024 3:50 PM (20 minutes)

Simulations based on particle methods, such as Smoothed Particle Hydrodynamics (SPH), are known to be computationally demanding, consisting on numerous interactions between locally-defined neighbors. Compared to other numerical methods, SPH is mesh-free, meaning that computations are not restrained to a fixed grid: particles, acting as interpolation nodes, are instead free to move across the entire domain, leading to additional challenges when dealing with parallel computations.

While such methods have for long been executed in parallel on multi-core CPUs, in recent years the increasing adoption of many-core accelerators, such as GPUs, has opened up the field of parallel computing to new possibilities.

However, parallel models and techniques do often differ between multi-core and many-core systems, requiring particular attention in coordinating the execution of threads and memory operations for the latter.

Moreover, hardware fragmentation and vendor-specific programming interfaces are still characterizing their market. Hence, support for various hardware configurations may easily lead to non-trivial and less maintainable implementations.

To leverage over those differences, some higher-level specifications have become available recently, such as the SYCL programming standard, which provides an interface for compiling ISO C++ code on various back-ends, including GPU APIs.

The following work highlights the initial effort in adopting the SYCL standard for the execution of SPHinXsys, an open-source multi-physics library. The result is an execution model able to run the same implementation on variable (heterogeneous) hardware, with considerable speed-up compared to the current multi-core CPU parallelization.

The discussion will primarily focus on the difference between multi-core and many-core parallelization, describing how the existing parallel methods have been adapted to be executable with SYCL. Among others, representation of data-structures for parallel access, communication strategies, and parallel methods for data sorting will be topics discussed in depth. Minimizing the effort for the user to adopt this new execution model has also been taken into consideration, reducing the changes required to port an existing simulation. Execution details are designed to be transparent to the library user, not requiring particular knowledge of the underlying execution model. Finally, benchmarks will be presented, showcasing performance comparisons between the current multi-core CPU implementation and the newly introduced SYCL parallelization with a GPU back-end.

Slot length

Primary author: GUARNIERI, Alberto (Technical University of Munich)

Co-author: HU, Xiangyu

Presenter: HU, Xiangyu

Session Classification: Parallelization and HPC Infrastructure

Track Classification: HPC and Massive Data: Parallelization and HPC Infrastructure