EGU22-11212
https://doi.org/10.5194/egusphere-egu22-11212
EGU General Assembly 2022

Lightweight embedded DSLs for geoscientific models.

Zbigniew Piotrowski[1], Daniel Caviedes Voullieme[1], Jaro Hokkanen[2], Stefan Kollet[1], and Olaf Stein[1]
[1]Forschungszentrum Jülich GmbH, Jülich, Germany
[2]CSC – IT Center for Science, Finland

On the map of the ESM research and operational software efforts, a notable area is occupied by the mid-size codes that benefit from established code design and user base and are developed by the domain scientists. Contrary to the major operational frameworks and newly established software projects, however, developers of such codes cannot easily benefit from novel solutions providing performance portability, nor have access to software engineering teams capable to perform full code rewrite, aiming at novel hardware architectures. While evolving accelerator programming paradigms like CUDA or OpenACC enable reasonably fast progress towards execution on heterogenous architectures, they do not offer universal portability and immediately impair code readability and maintainability. In this contribution we report on a lightweight embedded Domain Specific Language (eDSL) approach that enables legacy CPU codes to execute on GPU. In addition, it is minimally invasive and maximizes code readability and developer productivity. In the implementation, the eDSL serves as a front end for hardware dependent programming models, such as CUDA. In addition, performance portability can be achieved efficiently by implementing parallel execution and memory abstraction programming models, such as Kokkos as a backend. We evaluate the adaptation process and computational performance of the two established geophysical codes: the ParFlow hydrologic model written in C, and the Fortran-based dwarf encapsulating MPDATA transport algorithm. Performance portability is demonstrated in the case of ParFlow. We present scalability results on state-of-the-art AMD CPUs and NVIDIA GPUs of JUWELS booster supercomputer. We discuss the advantages and limitations of the proposed approach in the context of other direct and DSL-based strategies allowing for exploitation of the modern accelerator-based computing platforms.