



Contribution ID: 12

Type: **not specified**

Tasmania: A stencil-oriented framework for performance-portable weather and climate applications in Python

Tuesday, 21 September 2021 16:20 (20 minutes)

Dynamical cores and physical parameterizations have traditionally been engineered in isolation for the sake of tractability. The compartmentalization of both the modeling, numerical and software development facilitates the proliferation of model components with incompatible structures. The implementation is often tailored to the specifics of the intended host model, so that transferring schemes between models often requires a significant effort and involves sophisticated interfaces. In the attempt to alleviate these problematics, the Tasmania framework offers domain scientists a favorable platform in Python to write model-agnostic and plug-compatible components with a clean and common interface. Each component explicitly states the name, dimensions and units of both the fields required as input and the quantities provided in output. This enhances the overall readability of the code and accommodates sanity checks, units conversion, axes transposition, and data-dependency analyses. Components can be composed via couplers to form a hierarchy of models with different levels of complexity. Each coupler pursues a well-defined physics-dynamics or physics-physics coupling algorithm. Within each component, stencil-based computations arising from Eulerian-type dynamics and single-column physics can be encoded using a variety of tools, ranging from scientific computing packages like NumPy and CuPy, to just-in-time accelerators like Numba, to domain specific libraries like GT4Py. Indeed, Tasmania allows to define, organize and manage multiple backends in an organic fashion. Infrastructure code ensures that memory allocation, kernel compilation and stencil launch are properly dispatched. This highly relieves the user code of boilerplate and backend-specific instructions, favors a smooth transition from prototyping to production, and ultimately enables performance portability. By means of an idealized three-dimensional mountain-flow model, we show how Tasmania combined with GT4Py provides an effective solution to build modular applications which run optimally on both CPU and GPU, thus overcoming the well-known slowness of the Python interpreter.

Primary author: UBBIALI, Stefano (ETH Zurich)

Co-authors: Dr BIANCO, Mauro (Swiss National Supercomputing Center, Switzerland); Dr GONZALEZ PAREDES, Enrique (Swiss National Supercomputing Center, Switzerland); Prof. SCHÄR, Christoph (ETH Zurich, Switzerland); Dr SCHLEMMER, Linda (Deutscher Wetterdienst, Germany); Prof. SCHULTHESS, Thomas C. (Swiss National Supercomputing Center, Switzerland)

Presenter: UBBIALI, Stefano (ETH Zurich)

Session Classification: Session 4

Track Classification: 19th Workshop on high performance computing in meteorology