ACCORD

A Consortium for COnvection-scale modelling Research and Development

Preparing ACCORD codes for hybrid hardware architectures

Daan Degrauwe, Piet Termonia

ACCORD ASW 2022

(ever-changing) Context

- **Adaptation of NWP codes to heterogeneous hardware is a long-standing topic**

  (although ACCORD was not really very active in this)

- **Last year was characterized by increased efforts, mainly at ECMWF and MeteoFrance (see Michael's and Philippe's presentations), but also in some other ACCORD countries**

- **DestinE-Extremes may provide opportunities**

Steps to take

- **Refactoring, i.e. preparing the code to enable running on heterogeneous architectures**

- **Actual adaptation of code to accelerators like GPUs**

Refactoring

- **The flexibility of the code must be increased to make it adaptable to heterogeneous hardware**

- **Flexibility is closely related to cleanliness, e.g. pruning of unused options and variables, removing calculations from control routines, developing well-defined interfaces to parameterizations, etc.**

Refactoring

- **New smart (hardware-aware) data structures are being introduced (by MeteoFrance and ECMWF) in the gridpoint calculation control routines; these will enter in cy48t3.**

- **Task teams were formed to address the cleaning and refactoring for the HARMONIE-AROME and ALARO CSC's.**

Adaptation

Major challenges for the actual adaptation are:

- **ACCORD consists of 26 countries, each with their own (operational) HPC solution.**
- **Maintainability regarding new scientific developments: keep (as much as possible) a single code base**
- **Maintainability regarding future hardware developments**

Adaptation

For different parts of the code, different adaptation strategies are considered:

- **Spectral transforms: rely on hardware-optimized external libraries; e.g. ECMWF has made the global transform package publicly available, including a branch specific for NVIDIA GPUs.**
- **Gridpoint calculations: rely on code-transformation tools like Loki to generate hardware-specific code through loop-reordering, memory layout, etc.**
- **Large potential of machine-learning emulators for physics parameterizations**

Adaptation: experience with Loki for HARMONIE-AROME (Rolf@MetNo)

- Pretty steep learning curve.
- Hoist loops, reorder loops, reorder array dimensions, etc.
- Insert OpenAcc pragmas
- Find parameters and determine data passing
- Makes it possible to easily test out optimization options.
- Mostly possible to generate code without affecting scientists.

Importance of testing

Detailed testing is crucial during and after this work:

- **Refactoring has big impact on code**
- **Testing of various configurations**
- **Benchmarking on new hardware platforms**
- **Consider smaller tests (e.g. at level of individual parameterizations) for faster problem detection and easier porting and optimizing (also by non-NWP-experts)**

Thank you !

ACCORD
A Consortium for COnvection-scale modelling Research and Development