ACCORD

A Consortium for COnvection-scale modelling Research and Development

Topical session on code refactoring and adaptation

Daan Degrauwe

ACCORD All-Staff Workshop
Zalakaros, 3 April 2025

Outline

- Introduction (Daan)
- On source-to-source translation tools (Erwan)
- On FieldAPI (Judicaël)
- On APL_AROME refactoring (Rolf)
- On ALARO porting status (Daan)
- Discussion (everyone)

Reminder: GPU porting strategy

Following (and relying on!) developments by MeteoFrance and ECMWF

3 key ingredients

- usage of “smart” (i.e. GPU-aware) structures to handle data transfers between CPU and GPU
- Using source-to-source translation tools to generate GPU-code from CPU-code
- Using highly-optimized platform-specific libraries when available (e.g. FFT, BLAS)

Effect on code

Relatively big code changes in preparation of GPU porting of the model:

- updated coding norms to satisfy source-to-source translation tools
- encapsulation of data in FieldAPI structures
- restructuring of physics (APL_ARPEGE, APL_ALARO, APL_AROME)
- usage of fypp files to avoid boilerplate code
- trend towards externalising packages (e.g. fiat, field_api, ectrans, phyex)

… over to specialists on these topics

Status of porting ALARO to GPUs

Dynamics refactoring and porting

Dynamics (at least settings used in ALARO configuration) have been refactored and ported with (for now, fxtran) source-to-source translation

CPU

GPU

_PARALLEL routines

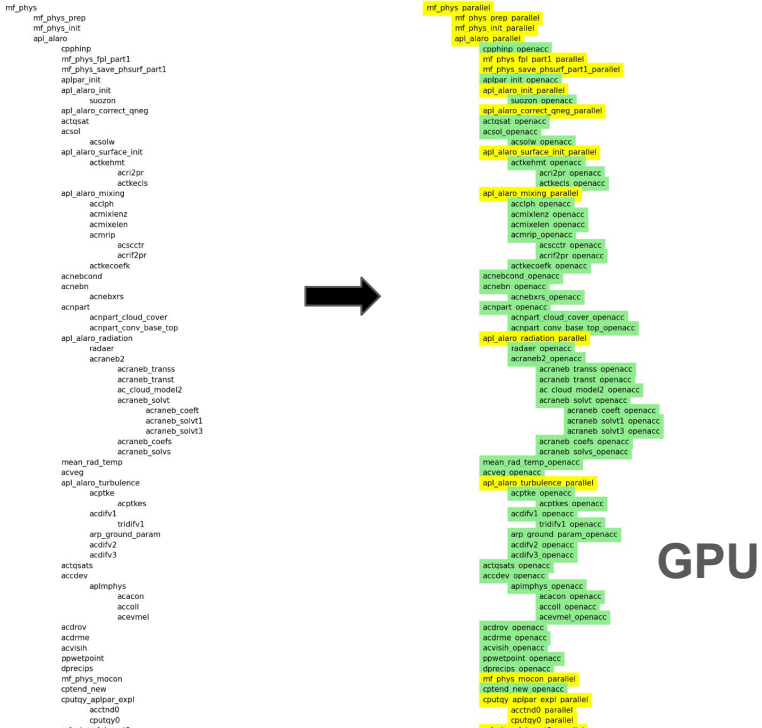
_OPENACC routines

Manually ported

ACCORD
A Consortium for COnvection-scale modelling Research and Development

ALARO physics refactoring and porting

ALARO physics have been refactored and ported with source-to-source translation



CPU

GPU

_PARALLEL routines

_OPENACC routines

Manually ported

ACCORD
A Consortium for COnvection-scale modelling Research and Development

Dynamics and ALARO physics (DEODE)					
Model component		Porting method	CPU run time	GPU(NVIDIA)	GPU(AMD)
Dynamical core	Spectral transform	Manual, OpenACC/OpenMP		Complete, **but under revision**	In **DEODE-protype**
	Gridpoint dynamics	FIELD API + Loki/Fxtran		Complete	To be tested
	Semi-Lagrangian	FIELD API + Loki/Fxtran		Complete	To be tested
	Spectral solver	FIELD API + Manual (for now)		Complete, **to be integrated**	
	LBCs			Complete	
Physics	ACRANEB2	FIELD API + Loki/Fxtran		Complete	In **DEODE-prototype**
	APLMPHYS	FIELD API + Loki/Fxtran		Complete	Stand-alone ready
	TOUCANS	FIELD API + Loki/Fxtran		Complete	
	3MT			Not needed for DEODE	Not needed for DEODE
	SURFEX				
EPS model	SPP (MF)				
Diagnostics/IO	DDH				
	IO (MF)				

Complete	Demonstrated	Working on it	External issues	Not started yet	Out of scope

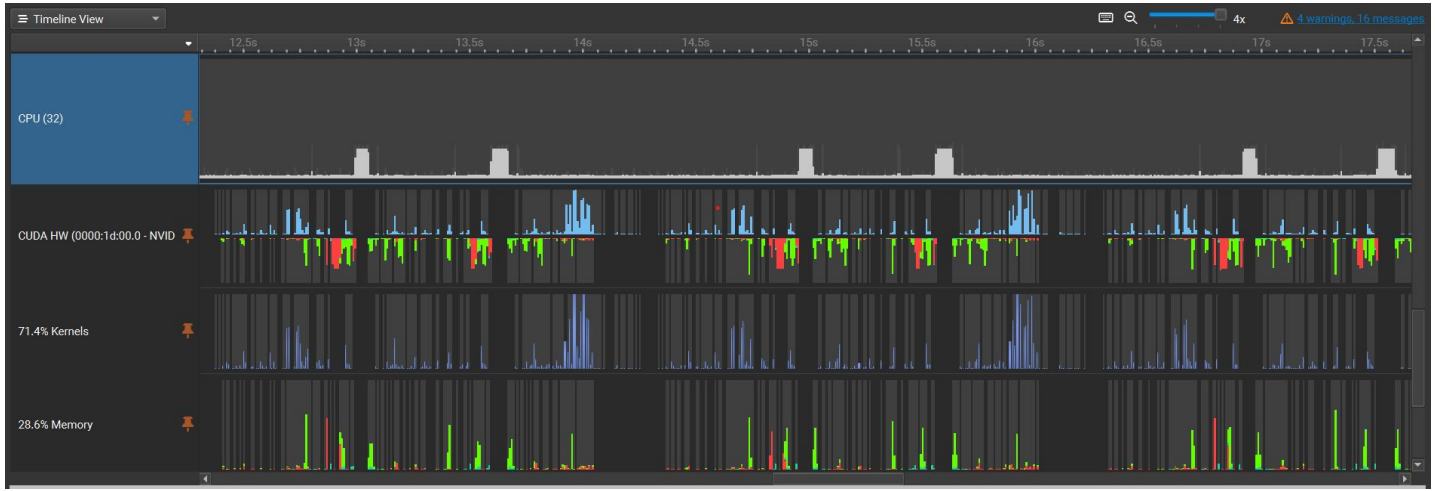
Example: ALARO run on LEONARDO HPC

Profile of a few timesteps (ALARO on a toy domain) on Leonardo HPC

CPU usage: mostly idle

GPU usage: quite busy

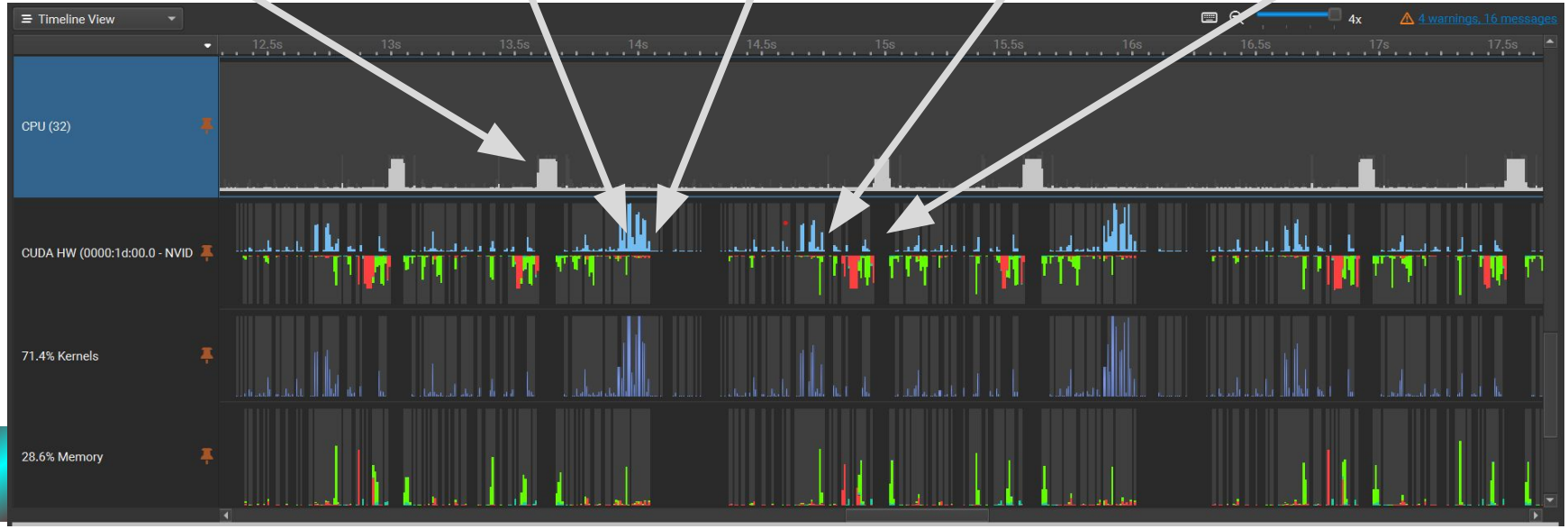
CPU-GPU transfers: To be optimized



ACCORD
A Consortium for COnvection-scale modelling Research and Development

Example: ALARO run on LEONARDO HPC

CPU	GPU	GPU	GPU	GPU
Spectral transforms + Helmholz solver	Radiation scheme	Turbulence scheme	Advection scheme	Lateral boundary conditions



Next steps

Further developments:

- Integration of refactored Semi-Implicit to have full ALARO timestep on GPU
- Porting of PHYEX parameterizations
- Refactoring/porting of other parts of the model: SPP, DDH, …?

Consolidation:

- (refactored gridpoint dynamics, semi-Lagrangian, LBCs, ALARO physics, (HARMONIE-)AROME physics entering CY50T1)
- (testing GPU build with ial-bundle/cmake: done)
- Integration of source-to-source translation in build system
- Documentation and training material

Discussion topics

- Code evolution (not only) due to refactoring: remarks, comments?
- Documentation and training on code refactoring: best format? Tutorial? Guidelines for developers? Webinar?
- Integration in build system (pending move from fxtran to loki?) and Davai
- …?