ACCORD

A Consortium for COnvection-scale modelling Research and Development

Progress in code adaptation to GPUs

Daan Degrauwe, on behalf of the SPTR contributors

ACCORD ASW 2024

Motivation (same as last year)

- Top 500 HPC systems:

  16 out of 20 top systems have accelerators

- Green 500:

  40 first systems have accelerators

- EuroHPC infrastructure is targeted in the DE_330 project

- Trend towards using external HPC facilities for research and even operations

- Infrastructure targeting AI/ML-applications is GPU-powered

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,194.00	1,679.82	22,703
2	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	4,742,808	585.34	1,059.33	24,687
3	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Microsoft Azure United States	1,123,200	561.20	846.84	
4	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
5	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,752,704	379.70	531.51	7,107
6	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, EVIDEN EuroHPC/CINECA Italy	1,824,768	238.70	304.47	7,404

Adaptation strategy

- Make adaptation as transparent as possible to science developers: principle of “separation of concerns”
- Make sure performance on CPUs is not impacted
- 3 pillars of code adaptation:
  - Smart (hardware-aware) data structures
  - Source-to-source translator tools
  - Hardware-specific libraries

Smart data structures: FIELD_API

- Developed by Météo-France and ECMWF, available on `https://github.com/ecmwf-ifs/field_api`

- Entered cy49t1

- Keep track of synchronization of fields on host (CPU) and device (GPU), performing (costly!) transfers only when necessary

- Brings flexibility in terms of which part of the model to run on which device

Smart data structures: FIELD_API

- Used at control layers, not inside low-level scientific code such as physics parameterizations

- Wrapping existing data structures in FIELD_API objects is a big effort
  - +/- finished for physics parameterizations, gridpoint dynamics, SL
  - not yet in diagnostics (e.g. DDH), LBC, spectral transforms

Restructuring of physics

- Separation between APL_ARPEGE (cy48t3) and APL_ALARO (cy49t1)
- Cleaning, restructuring and separation between control routines and computation routines for APL_AROME (cy49t1)
- Adhere to updated coding norms (ongoing!)
- Enable scripted conversion from CPU-targeted coarse granularity to GPU-targeted fine granularity.

Source-to-source translators

- General idea: automated (scripted) conversion of existing Fortran code targeting CPUs to Fortran code targeting GPUs
- Two tools currently used: fxtran+perl scripts (Météo-France), and loki (ECMWF). Transfer of existing fxtran recipes to loki is planned.
- Significant build-up of know-how on loki among ACCORD partners

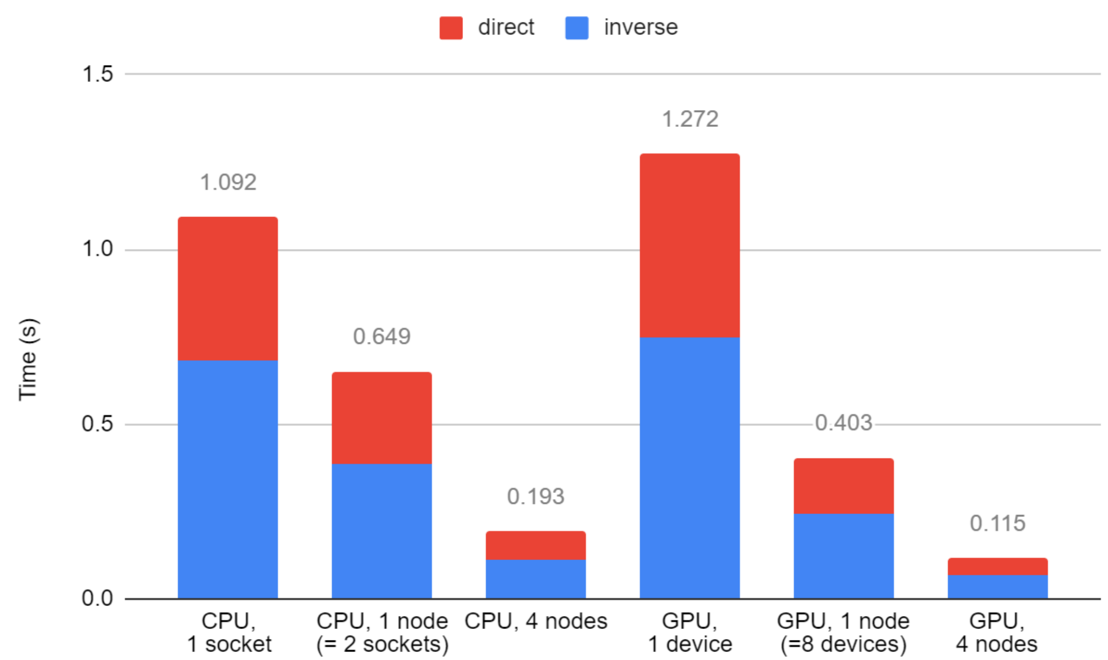
Porting of physics parameterizations

- Requires some changes to the code to enable source-to-source translation (updated coding norms!)
- Most ARPEGE parameterizations have been ported to GPUs
- ALARO radiation scheme ACRANEB2 and microphysics scheme APLMPHYS have been ported

Porting of spectral transforms

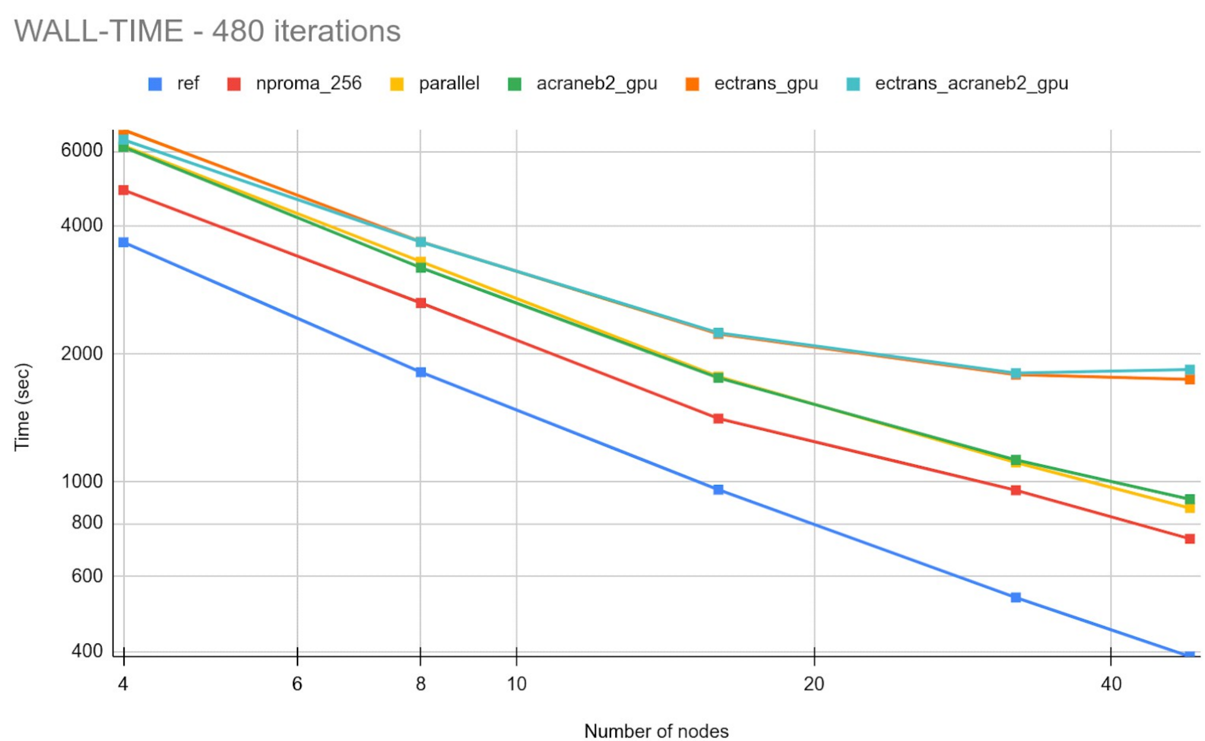
- Hardware-optimized libraries exist for FFTs: cuFFT for NVIDIA, rocFFT for AMD
- Computations inside spectral transforms (array transpositions, computation of derivatives) are ported with OpenACC/OpenMP

Performance on LUMI (AMD MI250X GPUs), including CPU-GPU transfers



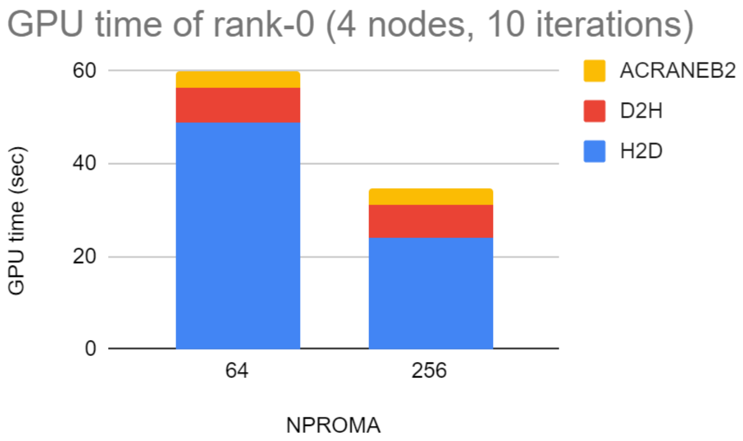
Putting the pieces together

- GPU-enabled ALARO forecast, with ACRANEB2 and spectral transforms on GPU
- Callable from deode-prototype scripting system
- Weak scalability tests on LUMI
- No direct GPU-GPU MPI communications (yet)



Putting the pieces together

- CPU-GPU data transfers kill performance



- Porting more parts of the model to GPU should solve this

Conclusion

- Developments have been ongoing for quite some time:
  - Code refactoring
  - Source-to-source translation tools
  - Smart data structures
  - Porting of small pieces of code

- Finally at a point where pieces come together: the GPU-enabled ALARO run is an important milestone

  … even though performance is not impressive (yet!)

What's next?

- Increase efforts on AROME and HARMONIE-AROME !
- Port more pieces to GPU to improve performance
- Consolidate developments, e.g. merge ACCORD developments on ectrans (LAM, AMD GPUs) with ECMWF's

Thank you!