Status of the COSMO - EULAG Gridtools port

Zbigniew Piotrowski, Marcin Polkowski, Adam Ryczkowski

Institute of Meteorology and Water Management - National Research Institute

September 12th, 2017

CEL-ACCEL work package 1 - Accelerator port of COSMO-EULAG

- Reformulation of EULAG diffusion package in Fortran to facilitate adaptation to accelerators
- GridTools port of COSMO-EULAG, including serialization against Fortran results with Serialbox
- Directives-based port of COSMO-EULAG(low priority)

Adapting COSMO-EULAG to accelerators using GridTools: strategy

- Advancing dual Fortran - GridTools C++ codebase (at least until GridTools reaches maturity)

Adapting COSMO-EULAG to accelerators using GridTools: strategy

- Advancing dual Fortran - GridTools C++ codebase (at least until GridTools reaches maturity)
- Major effort: refactoring and optimization of stencil components of EULAG dynamics; ongoing since 2014

Adapting COSMO-EULAG to accelerators using GridTools: strategy

- Advancing dual Fortran - GridTools C++ codebase (at least until GridTools reaches maturity)
- Major effort: refactoring and optimization of stencil components of EULAG dynamics; ongoing since 2014
- In parallel: establishing the development environment for the EULAG stencils ...

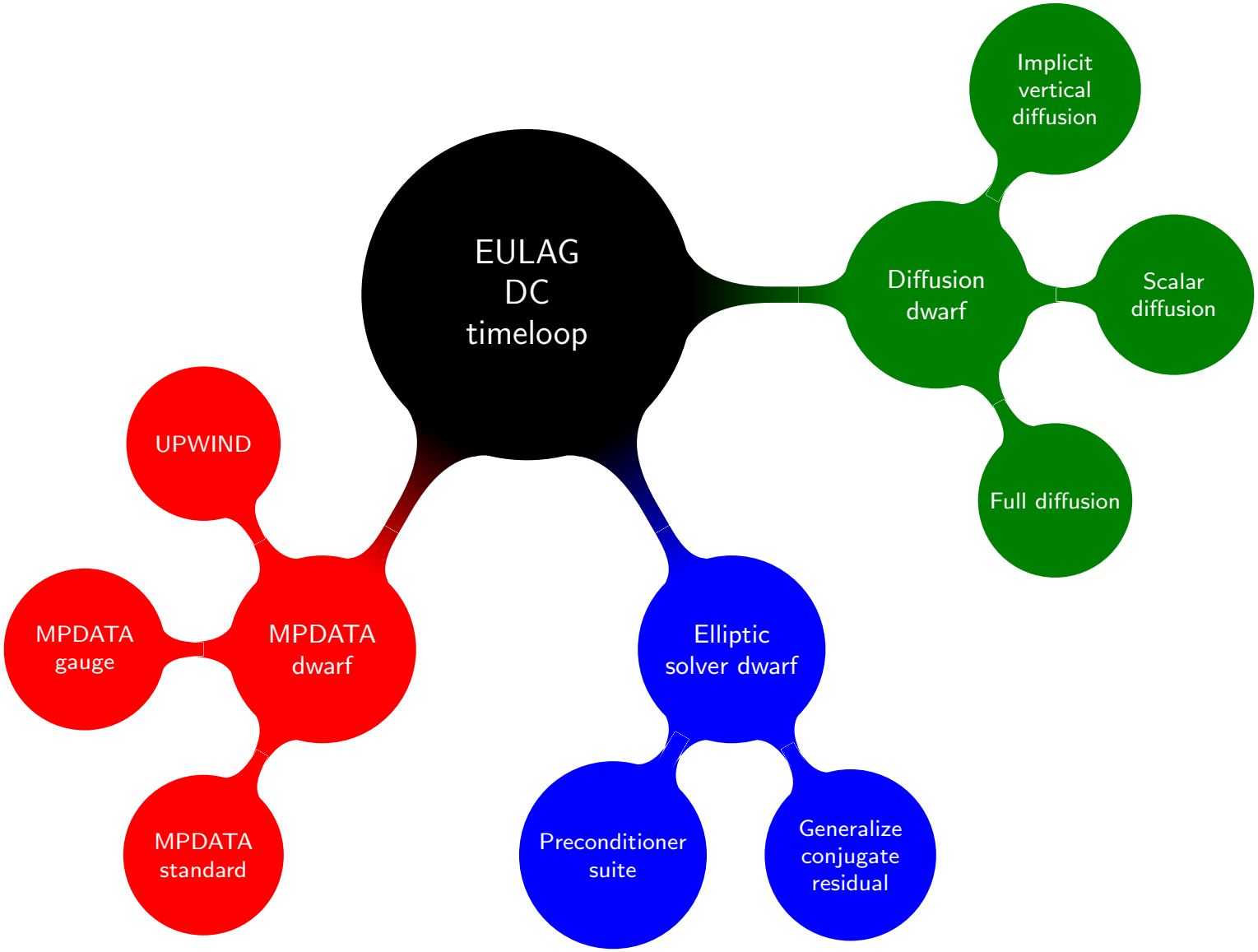
Adapting COSMO-EULAG to accelerators using GridTools: strategy

- Advancing dual Fortran - GridTools C++ codebase (at least until GridTools reaches maturity)
- Major effort: refactoring and optimization of stencil components of EULAG dynamics; ongoing since 2014
- In parallel: establishing the development environment for the EULAG stencils ...
- ... and porting the first EULAG algorithms into GridTools: UPWIND and one of the two MPDATA schemes

Adapting COSMO-EULAG to accelerators using GridTools: strategy

- Advancing dual Fortran - GridTools C++ codebase (at least until GridTools reaches maturity)
- Major effort: refactoring and optimization of stencil components of EULAG dynamics; ongoing since 2014
- In parallel: establishing the development environment for the EULAG stencils ...
- ... and porting the first EULAG algorithms into GridTools: UPWIND and one of the two MPDATA schemes
- Note that unlike for the RK core, the intermediate plain C++ dynamical core will not be created.

Envisaged COSMO-EULAG structure



Status of the COSMO-EULAG dynamics' dwarf implementations

Implicit elliptic(Helmholtz) solver was optimized with the advent of the COSMO-EULAG compressible. MPDATA advection has been optimized within the ESCAPE H2020 project. The optimization of EULAG 3D diffusion has begun within the CEL-ACCEL project with the implementation of the 3D scalar EULAG diffusion operator for general grids.

Dwarf	**MPDATA**	**Elliptic solver**	**Diffusion**
Dwarf (miniApp form)	Complete	Advanced	3D scalar diffusion
Refactoring status	Mature	Mature	Mature
GRIDTOOLS C++ port	Proof of concept	Just started	Just started

MPDATA and diffusion dwarf: facts

- Since regional NWP targets at limited number of cores, we focus on single core performance.

Advection and diffusion dwarfs: per-core performance improvement

Algorithm	1st order UPWIND	MPDATA standard	MPDATA gauge	3D diffusion
32x32x96 grid	1.58	1.40	2.33	2.82
16x16x128 grid	1.36	1.27	2.01	2.12

- Two times speedup is roughly consistent with earlier experiences with structured and unstructured elliptic solver and also with earlier code refactoring and C++ implementation performed on the operational Runge-Kutta dynamical core of COSMO.
- Size of the code is one-fourth of the legacy version.
- MPDATA and other EULAG dwarfs enjoy compact stencils and demand only single halo line in each direction.

Tools and technical efforts for GridTools implementation

The following tools were used and corresponding efforts were performed:

- Gitlab server at IMGW
- Modular cmake building environment for Fortran and C++ codes; including generation of the Eclipse projects
- Attempts to minimize efforts related to C++ GridTools complexity as for autumn 2016.
- Designing drivers for analytical test cases both in Fortran and C++ for the diffusion and advection miniapps
- Designing the testing framework and CI
- Implementing the dwarfs using the Gridtools library

Experiences from GridTools implementations

Gridtools port of the MPDATA gauge and UPWIND schemes has just been finished (A. Ryczkowski, PROPOZE project efforts). The development revealed the necessity of extension (or evaluation of the existing solutions) to GridTools to accommodate new types of boundary conditions. The code is not ready for optimizations due to the temporary treatment of the lateral boundary conditions.
Main issues in the implementation:

- Cifficulty to diagnose the actual state of computations, resulting from "take all or nothing" state of the verification framework, together with lack of the possibility of debugging
- Corresponding recommendation: striving for the gradual rather than radical approach and provide standard interfaces/technologies whenever possible (e.g. Netcdf output instead of native format of the serialization software)
- Replication of the boilerplate code is an issue, but it is of secondary importance as compared to the executing limited confidence and having control on the computation process together with having access to the base of examples/documentation

Experiences from GridTools implementations

Issues in the implementation, continued:

- Embarking on the GridTools framework with the large-scale project seems to be significant software-engineering task.
- Corresponding recommendation: providing sample git repository containing all necessary components/libraries/third party libraries with the basic examples. Such repository should address major areas of modern software engineering.
- Boundary conditions are the key for efficient integrations over complex topography; there is a need for the standarization of the typical realization of boundary conditions.
- Introduction of the Fixture Class of Google Test seem to be not well justified at first sight.
- GridTools (Verification) learning curve without good documentation seems just too steep if one is to shy to constantly ask for help.

Conclusions

- Before striving for the optimal performance, there has to be well-constructed numerical solver that has also to be well coded (unless developing from scratch in DSL, which we may see in the future)
- DSL community has to decide on the level of accessability of the DSL; should it really be for dummies at this moment and for what price ? I don't see any issues with the use of DSL if the above condition is fulfilled.
- Smaller/poorer countries should make a decision on implementation of the GPU dycore based on economical principles, which seems to be more important that the strategic control over everything.
- In the context of EULAG dynamical core, the GridTools development translates directly to the capabilities of the EULAG standalone solver. Here, the capability for 'faster' runs on GPU can give a boost, e.g. for the MHD solar climate

Let's enjoy the trip !

From Jeremiah, 31:
15 Yahweh says this: A voice is heard in Ramah, lamenting and weeping bitterly: it is Rachel weeping for her children, refusing to be comforted for her children, because they are no more.
16 Yahweh says this: Stop your lamenting dry your eyes, for your labour will have a reward, Yahweh declares, and they will return from the enemy's country.