**Advanced Computation and I/O Methods for Earth-System Simulations**

Julian M. Kunkel, Thomas Ludwig, Thomas Dubos, Naoya Martyama, Takayuki Aoki, Günther Zangl, Hisashi Yashiro, Ryuji Yoshida, Hirofumi Tomita, Masaki Satoh, Yann Meurdesoif, Nabeel Jum’aah, Anastasia Novikova (Contact: kunkel@dkrz.de)

---

### Motivation
- Several groups work on icosahedral-grid based climate/weather models
- (Difficulties for Ensemble simulations - too fast or too slow scale)
- Code is very complex and difficult to react
- Climate prediction creates huge data volumes

### Limitations of general-purpose programming languages
- Semantic and syntax restrict programmers productivity
- Performance is hardly portable between architectures

### Existing Domain-Specific Languages
- May create optimized code for different architectures
- Technical languages with limited relation to scientific domain
- Typically require language-specific paradigm shift for scientists
- Unclear future of the framework/tool

### Existing scientific file formats
- Metadata for icosahedral data is not standardized
- Difficult to achieve good performance
- Pre-defined compression schemes achieve suboptimal ratio

---

### Scientific Work Packages: Objectives and Tasks

**WP 1: Towards higher-level code design**
- Foster separation of concerns: Domain scientists, scientific programmer and computer scientists
  - High level of abstraction, reflects domain science concepts
  - Independence of hardware-specific features, e.g. memory-layout
  - Convertible into existing languages and DSLs

**WP 2: Massive I/O**
- 1) Optimize file code for icosahedral data
- 2) Data reduction concepts
- 3) API for user-defined variable accuracy
- 4) Identifying required variable accuracy
- 5) Lossy compression

**WP 3: Evaluation**
- 1) Selection of representative test cases
- 2) Extraction of simple kernels
- 3) Common benchmark package/mini-IGCM
- 4) Benefit of the DSL for kernels/mini-IGCMs
- 5) Estimating benefit for full-featured models
- 6) I/O advances for full models

---

### Goals
- Enhance programmability and performance-portability
- Overcome storage limitations
- Provide a common benchmark for icosahedral models

---

### Collaboration
**Funded partners**
- Thomas Ludwig (University Hamburg)
- Thomas Dubos (Institut Pierre-Simon Laplace)
- Naoya Martyama (BIREN)
- Takayuki Aoki (Taisei Institute of Technology)

**Collaboration partners**
- DWD (I/O, DSL)
- DWD (ICON, I/O, DSL)
- University of Exeter (Matematicas aspects in the DSL)
- NCAS (GPL/ICON, GRIDTools compression)
- Intel (DSL backend optimizations for XeonPhi, CPU)
- NVIDIA (DSL-backend optimization for GPU)
- The HDF Group (I/O, abstracted data, compiler)
- NCAR (ISDP developers, first icosahedral model)
- Bull
- Univ
- Information exchange, participation in workshops

---

### Models
- **ICON**
- **DYNAMICO**
- **NICAM**

---

### DSL Tools
**Source-to-source translation**
- Translates GIDDSL code into
  - Architecture-optimized code (Off intermediate-language)
  - Light-weight easily maintainable translation tool, shipped with code
  - Integratable into Build-Systems
  - Offers a configurable translation procedure
  - Internal prototype with limited functionality is complete

**Compilations profiling**
- Learn optimal compilation options for each repository file
- Minimize time of repository build while keeping code performance

---

### GGDM Domain-Specific Language
- **GGDM:** General Grid Definition and Manipulation Language
- Abstracted scientific-domain based constructs for
  - Data types reflecting "grid" concepts
  - Variable Declaration & allocation on cells, edges and vertices
  - Iterators to traverse and update variables
  - Named neighborhoods in (truncated/hemispherical) grid
  - Developed in co-design with domain scientists

---

### Source code (dynamic) and GGDM version
- **GGDM version of the code above**
- **Source code**
- **GGDM version of the code above**

---

### Compression
- Development of Scientific Compression Library
  - https://github.com/juliankunkel/scl
- Users define the required accuracy
- In terms of relative/absolute/precision ...
- In terms of required performance
- The library picks a fitting algorithm
- Integration into HDF5 / NetCDF4

---

### Extending NICAM with a high-level framework
- GridTools
  - C++ template framework for weather and climate models
  - Architecture-independent programming interface for performance and portability
- Evaluating GridTools as a programming framework for NICAM
- Successfully ported representative NICAM stencil kernels with comparable performance as hand-tuned implementations

---

### Acknowledgement
This work was supported by the German Research Foundation (DFG) through the Priority Programme 1549: Software for Exascale Computing (SPPEXA).

---

https://wr.informatik.uni-hamburg.de/research/projects/aimes/start