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

Julian M. Kunkel, Thomas Ludwig, Thomas Dubos, Naoya Maruyama, Takayuki Aoki, Günther Zängl, Hisashi Yashiro, Ryuji Yoshida, Hirofumi Tomita, Masaki Satoh, Yann Meurdesoif, Nabeeh Jum'ah, Anastasiia Novikova

Scientific Computing
Department of Informatics
University of Hamburg

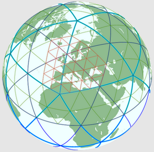
2017-03-21

Goals

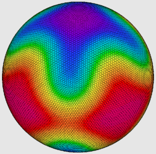
Address key issues of icosahedral earth-system models

- Enhance programmability and performance-portability
- Overcome storage limitations
- Shared benchmark for these models

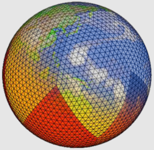
Covered models



ICON



DYNAMICO



NICAM

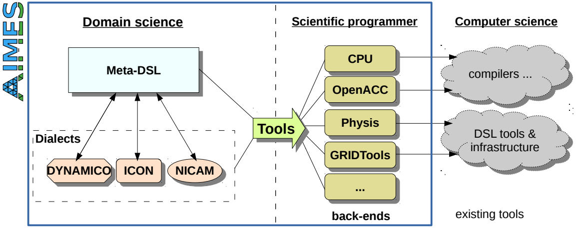
WP1: Towards Higher-Level Code Design

Recap: Goals of the WP

- Bypass shortcomings of general-purpose languages
  - Offer performance-portability
  - Enhance source repositories maintainability
  - Get rid of complexity in optimized-code development
  - Enhance code readability and scientists productivity
- Extend modelling programming language
  - Based on domain science concepts
  - Free of lower level details (e.g., architecture, memory layout)

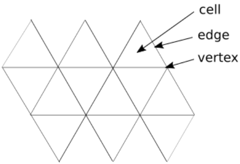
Approach

- Foster separation of concern
  - Domain scientists develop domain logic in source code
  - Scientific programmers write hardware configurations
- Source code written with extended language
  - Closer to domain scientists logic
  - Scientists do not need to learn optimizations
  - Write code once, get performance for various configurations
- Hardware configurations define software performance
  - Written by programmers with more experience in platform
  - Comprise information on target run environment

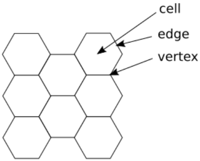


DSL Development

- Co-design with scientists to develop DSL constructs
  - Current version represents several iterations
  - GGDML: *General grid definition and manipulation language*
  - Grid definition
  - Grid-bound variable declaration
  - Grid-bound variable access/update
  - Stencil operations
- Hides memory locations and access details, data iteration
- Abstract higher concepts of grids, hiding connectivity details



a) Triangular grid



b) Hexagonal grid

Fortran vs. GGDML Code Example

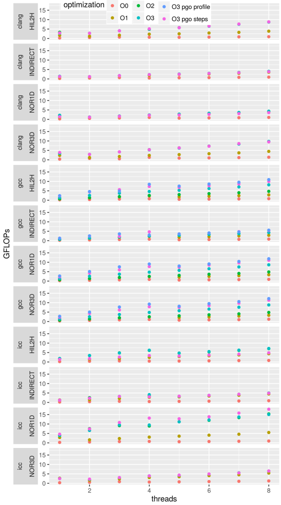
```
DO l=ll_begin,ll_end
!DIR$ SIMD
  DO ij=ij_begin,ij_end
   berni(ij,l) = .5*(geopot(ij,l)+geopot(ij,l+1)) +
       1/(4*Ai(ij)) *
       (le(ij+u_right)*de(ij+u_right)*u(ij+u_right,l)**2 &
       +le(ij+u_rup)  *de(ij+u_rup)  *u(ij+u_rup,l)**2   &
       +le(ij+u_lup)  *de(ij+u_lup)  *u(ij+u_lup,l)**2   &
       +le(ij+u_left) *de(ij+u_left) *u(ij+u_left,l)**2  &
       +le(ij+u_ldown)*de(ij+u_ldown)*u(ij+u_ldown,l)**2 &
       +le(ij+u_rdown)*de(ij+u_rdown)*u(ij+u_rdown,l)**2 )
  ENDDO
ENDDO
```

GGDML version of the code above

```
FOREACH cell IN grid
 berni(cell) = .5*(geopot(cell)+geopot(cell%above)) +
     1/(4*Ai(cell)) * REDUCE(+,N, le(cell%neighbour(N))*
     de(cell%neighbour(N))* u(cell%neighbour(N))**2)
END FOREACH
```

Status of Tools & Performance Analysis Experiments

- Initial tools developed for
  - Source-to-source translation
  - Flexible: adjustable DSLs
  - Performance analysis
  - Integration into build systems
- Explored opt. of mem. layout
  - 3D and 1D transformation
  - Hilbert filling curves & HEVI
  - With various compilers
    - Intel, GCC, CLang
  - Best layout depends on compiler!
- NICAM-DC ported to GridTools
  - Potential back-end for GGDML
- Hybrid Fortran for ASUCA weather
  - Code generation of CUDA

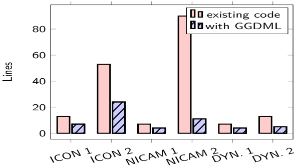


GGDML Impact on the Source Code

The DSL reduces development and maintenance effort

- LOC statistics

Model, kernel	lines (LOC) before DSL	lines (LOC) with DSL	words before DSL	words with DSL	characters before DSL	characters with DSL
ICON 1	13	7	238	174	317	258
ICON 2	53	24	163	83	2002	916
NICAM 1	7	4	40	27	76	86
NICAM 2	90	11	344	53	1487	363
DYNAMICO 1	7	4	96	73	137	150
DYNAMICO 2	13	5	30	20	402	218
total	183	55	911	430	4421	1991
relative size with dsl	30%		47%		45%	



- Predicting saving applying the DSL to 300k code of ICON
  - 100k infrastructure (does not change with the DSL)
  - Remaining code reduced according to our test kernels
  - COCOMO estimations

Software project	Version	Effort Applied	Dev. Time (months)	People require	dev. costs (M€)
Semi-detached		2462	38.5	64	12.3
	DSL	1133	29.3	39	5.7
Organic		1295	38.1	34	6.5
	DSL	625	28.9	22	3.1

WP2: Massive I/O

Recap: Goals of the WP2

- Optimization of I/O middleware for icosahedral data
  - Throughput, metadata handling
- Design of domain-specific compression (ratio $> 10 : 1$)
  - Investigate metrics allowing to define accuracy per variable
  - Design user-interfaces for specifying accuracy
  - Develop a methodology for identifying the required accuracy
  - Implement compression schemes exploiting this knowledge

WP2: Supported Quantities

Quantities defining the residual (error):

absolute tolerance: compressed can become true value ± absolute tolerance

relative tolerance: percentage the compressed value can deviate from true value

relative error finest tolerance: value definining the absolute tolerable error for relative compression for values around 0

significant digits: number of significant decimal digits

significant bits: number of significant decimals in bits

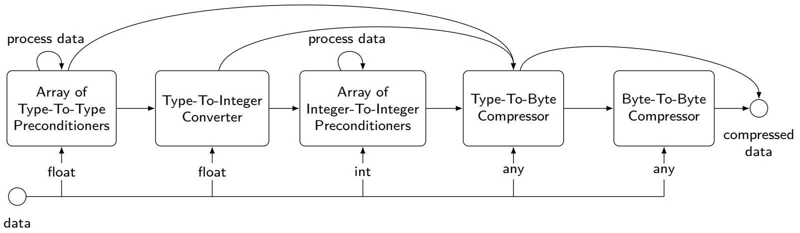
field conservation: limits the sum (mean) of field's change

Quantities defining the performance behavior:

absolute throughput: in MiB or GiB, or relative to network or storage speed

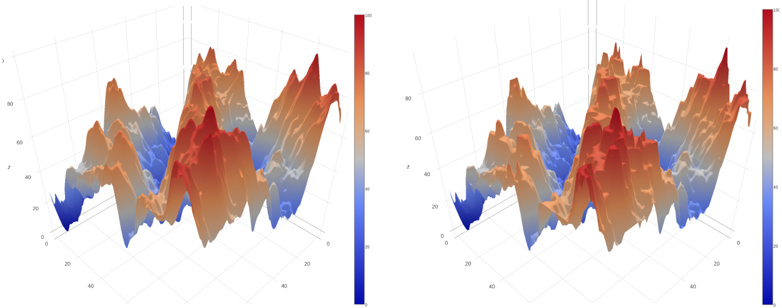
Architecture of SCIL

- Contains tools to
  - Create random patterns, compress/decompress, add noise, plot
- HDF5 and NetCDF4 integration; tools support NetCDF3, CSV
- Library with
  - Automatic algorithm selection (under development)
  - Flexible compression chain:



WP2: Example Synthetic Data

Simplex (options 206, 2D: 100x100 points)



Right picture compressed with Sigbits 3bits (ratio 11.3:1)

Analyzing Performance of Lossy Compression using SCIL

An initial experiment is conducted with SCIL

Data

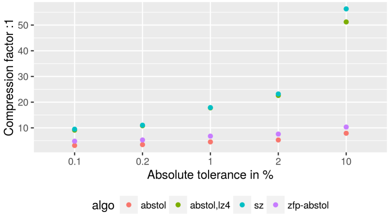
- Single precision (1+8+23 bits)
- Synthetic, generated by SCIL's pattern lib.
  - e.g., Random, Steps, Sinus, Simplex
- Data of the variables created by ECHAM
  - The climate model creates up to 123 vars

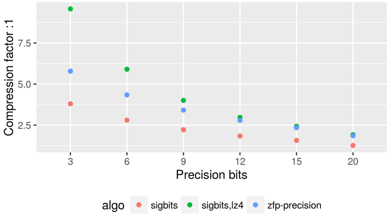
Experiments

- Single thread, 10 repeats
- Lossless (memcopy and lz4)
- Lossy compression with significant bits (zfp, sigbits, sigbits+lz4)
- Lossy compression with absolute tolerance (zfp, sz, abstol, abstol+lz4)
  - Tolerance: 10%, 2%, 1%, 0.2%, 0.1% of the data maximum value

WP2: Tolerance-Based Results

- Mean compression factor across all scientific files
- Factor 50:1 means space is reduced to 2% of the original size
- Note that SZ and ZFP do not always reach the set precision
  - Often the absolute and precision bit tolerance cannot be met





WP2: Comparing Algorithms for the Scientific Files

		Throughput [MiB/s]	
Algorithm	Ratio	Compr.	Decomp.
sigbits	0.45	464.4	620.7
sigbits,lz4	0.23	401.6	585.2
zfp	0.30	194.8	418.3

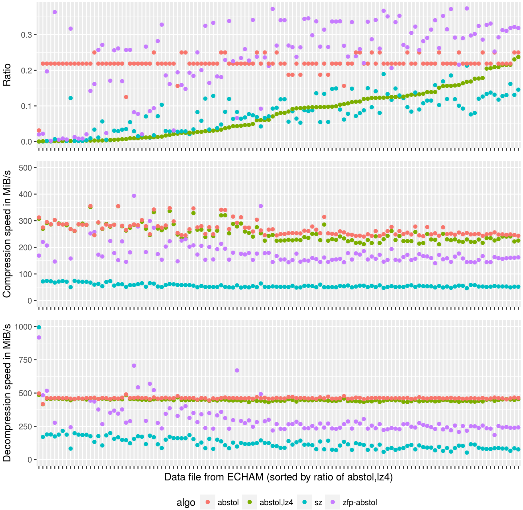
Table: Preserving 9 precision bits (instead of 23 from float) ≤ 0.56

		Throughput [MiB/s]	
Algorithm	Ratio	Compr.	Decomp.
abstol	0.22	269.3	461.7
abstol,lz4	0.08	253.4	446.8
sz	0.08	66.0	127.4
zfp	0.24	263.9	492.9

Table: For absolute tolerance with 1% of max value < 0.22

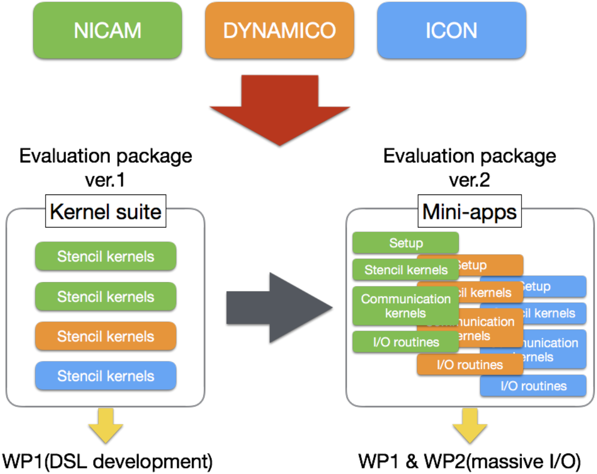
WP2: Results for Absolute Tolerance

Comparing algorithms using an absolute tolerance of 1% of the maximum value



WP 3: Evaluation

- Evaluating the DSL and domain-specific I/O advancements
- Providing a common benchmark package for all models



Kernel Extraction

- Extracted kernels (e.g. from NICAM)
  - 2-D(horizontal) diffusion: simple stencil
  - 1-D(vertical) tridiagonal matrix solver: with (i*j,k) array, recurrence k-axis
  - 3-D divergence damping: simple but large memory footprint
  - 2-D(horizontal) flux calculation with remapping: large memory footprint
  - 2-D(horizontal) flux limiter for tracer transport: complex, max()/min()
  - 1-D(vertical) flux limiter for tracer transport: complex, max()/min()

  - Setup routine for the coefficients of the stencil operators
  - Communication kernel

Summary

- AIMES covers programmability issues on the high-level
  - DSL-extensions enrich existing languages
  - Fosters separation of concerns, increase performance portability
  - We have reached a first concensus between scientists
  - Will work on further examples and (next) revision for the DSL
  - An initial prototype has been developed
  - Several backends have been explored
- AIMES addresses domain-specific lossy compression
  - (Help) scientists to define the variable accuracy
  - Exploit this knowledge in the compression scheme
  - Novel schemes compete with exististing algorithms
    - The choosing algorithm should always pick the best

Next Steps

- Bring together individual work on DSL backends
  - GridTools is a good candidate for a backend
- Refine project plan for next 12 month
- Explore use of further potential, e.g., YASK
- Provide shared DSL conventions and survey more scientists
- Extract and convert mini-apps of models to DSL
- Finalize SCIL API and interfaces to NetCDF/HDF5
- Integrate algorithm covering all accuracy quantities

Backup

Backup

Differences among three icosahedral atmospheric models

- Horizontal grid system
  - NICAM: co-located, semi-structured
  - DYNAMICO: staggered, semi-structured
  - ICON: staggered, unstructured

  - semi-structured means... "structured for stencil operation, unstructured for communication topology"

Partners and Expertise

Funded partners

- Thomas Ludwig (Universität Hamburg)
  I/O middleware, compression, ICON DSL
- Thomas Dubos (Institut Pierre Simon Laplace)
  Application I/O servers, compression, DYNAMICO
- Naoya Maruyama (RIKEN)
  DSL (Physis), GPUs, NICAM
- Takayuki Aoki (Tokio Institute of Technology)
  DSL (HybridFortran), language extension, peta-scale apps

Cooperation Partners

- DKRZ *(I/O, DSL)*
- DWD *(ICON, DSL, I/O)*
- University of Exeter *(Math. aspects in the DSL)*
- CSCS *(GPU/ICON, GRIDTool, compression)*
- Intel *(DSL-backend optimization for XeonPhi, CPU)*
- NVIDIA *(DSL-backend optimization for GPU)*
- The HDF Group *(I/O, unstructured data, compression)*
- NCAR (MPAS developers, another icosahedral model)
- Bull
- Cray

Information exchange, participate in workshops, [hardware access]