AIMES Kickoff Meeting

Julian M. Kunkel

Scientific Computing
Department of Informatics
University of Hamburg

2016-01-27

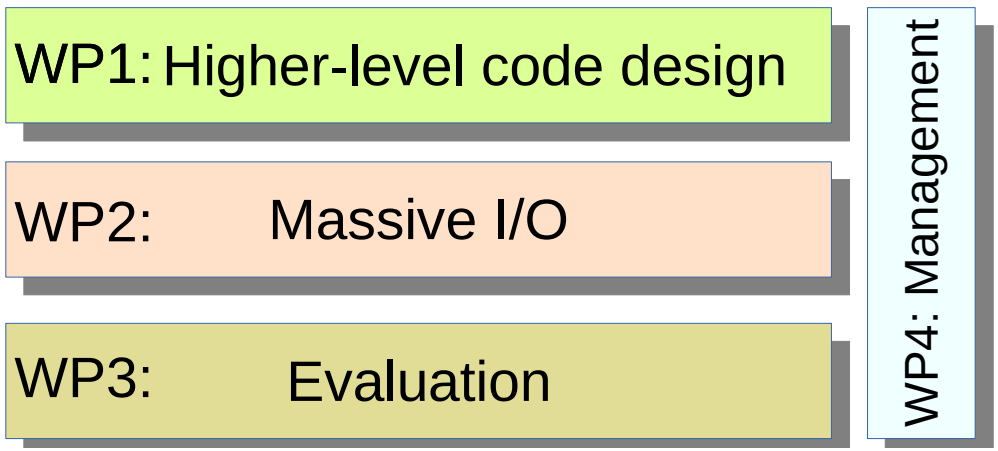
Agenda

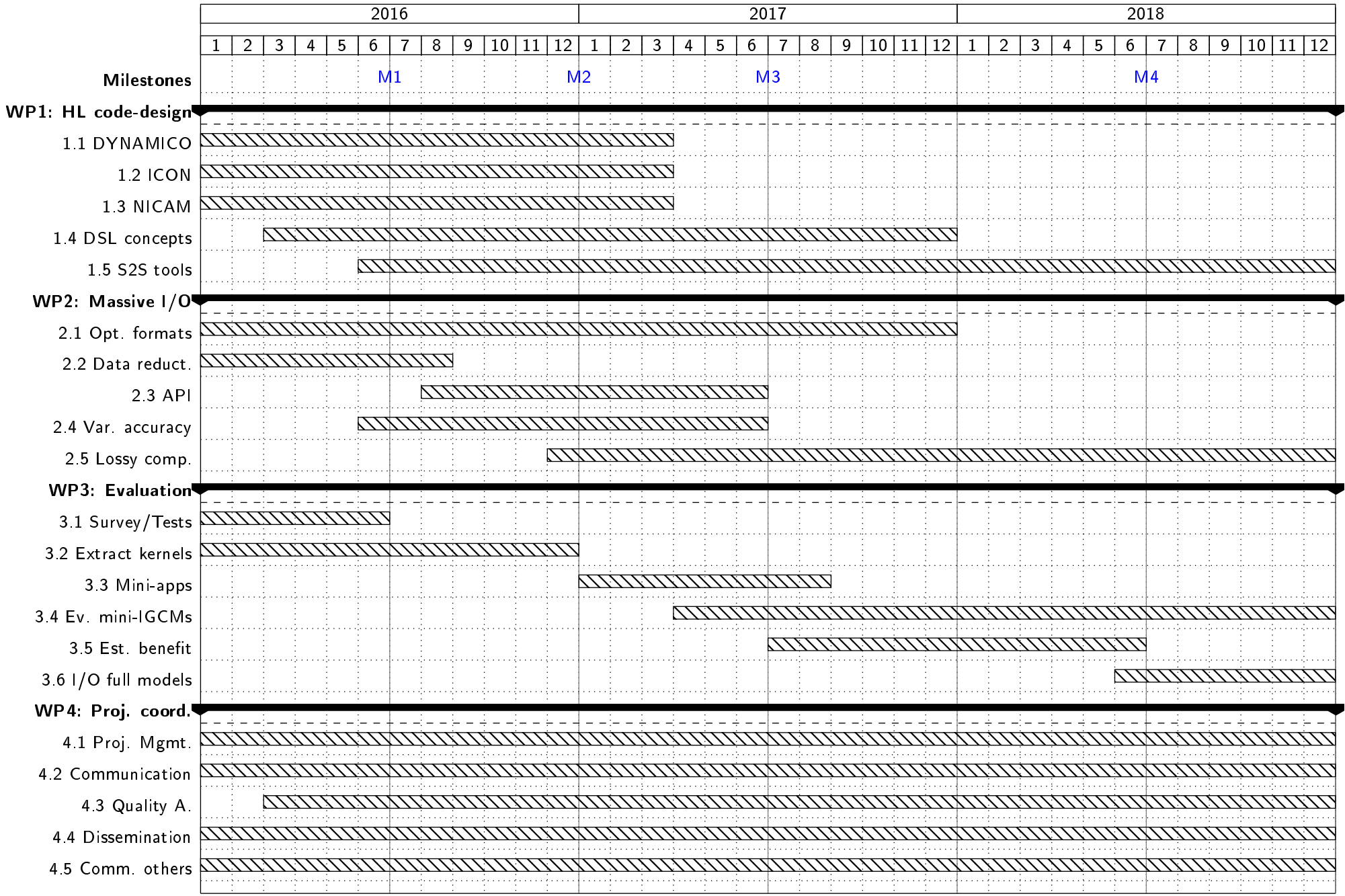
- 8:30 Welcome: Julian
- 8:40 Modell overview
  - 8:40 NICAM: Yashiro
  - 9:10 ICON: Günther Zängl
  - 9:40 DYNAMICO: Thomas Dubos
- 10:10 Coffee break (and selection of lunch)
- 10:40 Work packages (ongoing work, issues, 2016)
  - 10:40 WP1 DSL: Julian
  - 11:20 WP2 I/O: Thomas Dubos
  - 12:00 WP3 Evaluation: Yashiro
- 12:40 We have ordered lunch to this place
  WP4 (Administratives): Julian
- 13:00 Final discussions: all
- 13:45 *End*

Goals of this Meeting

- Clarify project content (and adjust it if necessary)
- Clarify (re-discuss) responsibilities and means for collaboration
- Discuss and identify joint strategy to achieve the project goals
  - Ensure that we have a plan to implement the goals
  - Especially for 2016

Work-Package Overview




2016
2017
2018
1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12
Milestones
M1
M2
M3
M4
WP1: HL code-design
1.1 DYNAMICO
1.2 ICON
1.3 NICAM
1.4 DSL concepts
1.5 S2S tools
WP2: Massive I/O
2.1 Opt. formats
2.2 Data reduct.
2.3 API
2.4 Var. accuracy
2.5 Lossy comp.
WP3: Evaluation
3.1 Survey/Tests
3.2 Extract kernels
3.3 Mini-apps
3.4 Ev. mini-IGCMs
3.5 Est. benefit
3.6 I/O full models
WP4: Proj. coord.
4.1 Proj. Mgmt.
4.2 Communication
4.3 Quality A.
4.4 Dissemination
4.5 Comm. others

WP1: Towards Higher-Level Code Design & Related Recent Work of UHAM

Julian M. Kunkel

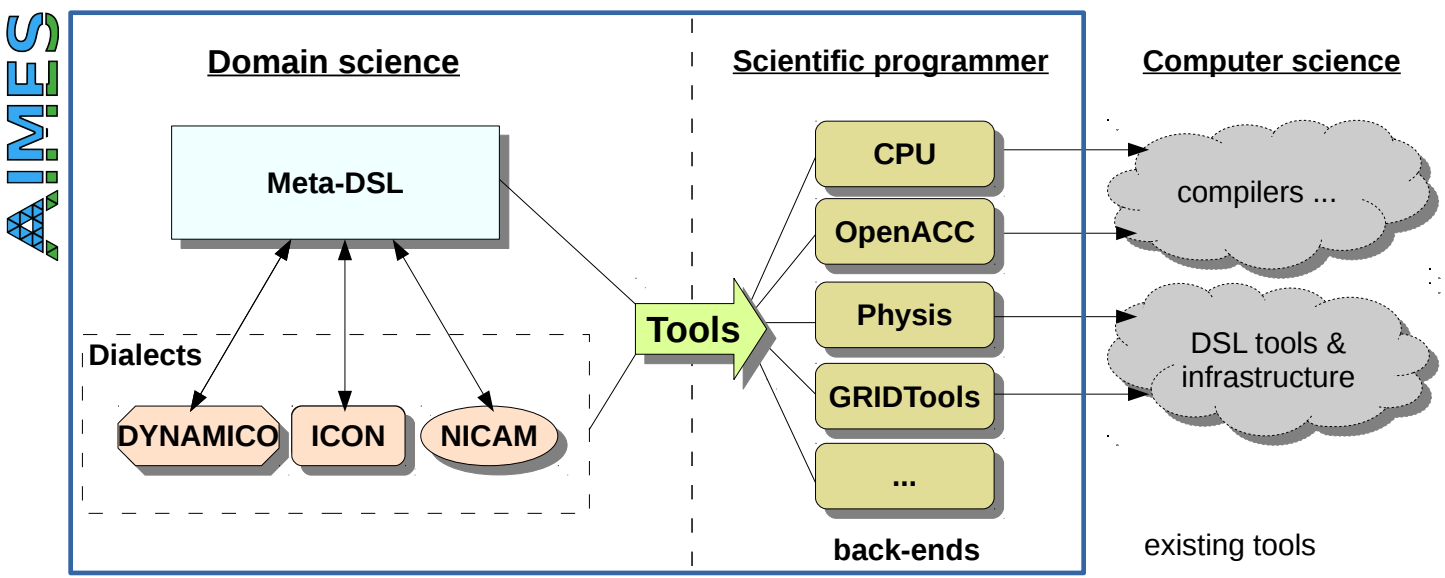
Scientific Computing
Department of Informatics
University of Hamburg

2016-01-27

Outline

Objectives: WP 1 Towards higher-level code design

- Separation of concerns: Domain science, scientific programmer, CS
  - High level of abstraction, close to application domain
  - Independence of hardware-specific features, e.g. memory-layout
  - Convertible into existing languages and DSLs



WP 1: Tasks & Strategy

- Develop and reformulate key parts of models into DSL-dialects
  - 1.1 DYNAMICO
  - 1.2 ICON
  - 1.3 NICAM
- 1.4 Design a common DSL concepts for all ICO models
  - Identify chances for common convention
  - But still allow for individual adjustments/formulations
- 1.5 Develop a source-to-source translation tool and mappings
  - Lightweight to allow scientists not to worry about maintenance
  - Language can be adjusted for individual models / groups

Risks and Mitigation Strategies

- Unavailability of runnable model test-code (WP 3)
  - We need to start ASAP to exchange code!
- Little uptake of the DSL concepts by the users
  - We need to involve the user communities
  - Can we identify a small number of key users to discuss DSL concepts/changed code?
  - Start with an incremental approach for the development
- Developed tool is not versataile
  - Create a repository of source/target examples (specimen repo)
  - Could be architecture-specific optimizations or target libraries (e.g. GridTools, ...)
- Abstraction does not allow architecture-specific optimizations (or into other target back-ends)
  - Test-driven development on specimen
  - Co-design between language wishlist and tool development

Prototypical Work for ICON

Dialect for ICON

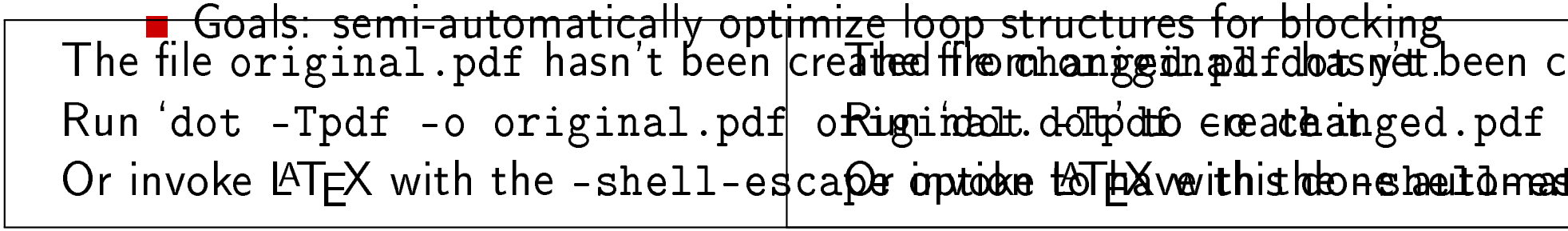
```
Subset                      :: p_patch
Edge                        :: edge
REAL(wp), Var1DCell         :: z_thermal_exp
foreach cell in p_patch (elevation rl_start to rl_end)
  z_thermal_exp(cell) = z_thermal_exp(cell) + &
    cvd_o_rd * p_nh%diag%ddt_exner_phy(cell) /  &
    (p_nh%prog(nnow)%exner(cell)* p_nh%metrics%inv_ddqz_z_full(cell))
end foreach
```

Original code == Translated code for CPUs

```
TYPE(t_patch), TARGET, INTENT(IN) :: p_patch
REAL(wp):: z_thermal_exp (nproma,p_patch%nblks_c)
i_startblk = p_patch%cells%start_block(rl_start)
i_endblk   = p_patch%cells%end_block(rl_end)
DO jb = i_startblk, i_endblk
  CALL get_indices_c(p_patch, jb, i_startblk, i_endblk,&
       i_startidx, i_endidx, rl_start, rl_end)
  DO jk = 1, nlev
    DO jc = i_startidx, i_endidx
      z_thermal_exp(jc,jb) = z_thermal_exp(jc,jb) +     &
         cvd_o_rd * p_nh%diag%ddt_exner_phy(jc,jk,jb)/ &
         (p_nh%prog(nnow)%exner(jc,jk,jb)*             &
         p_nh%metrics%inv_ddqz_z_full(jc,jk,jb))
    ENDDO
  ENDDO
ENDDO
```

Ongoing Work: Bachelor's Thesis

- Investigate high-level code modifications
  - Pull-up and push down of (if) conditions
  - Loop fusioning
  - Function merging (inlining)
  - Build a DAG for dependencies (could be exploited in other frameworks)
- Goals: semi-automatically optimize loop structures for blocking



Work Plan

Deliverables

- D1.1 Report: Model-specific dialect formulations – M15
- D1.2 Report and whitepaper: DSL concepts for icosahedral models — M24
- D1.3 Report and code: Advanced source-to-source translation tools — M36

Dependencies on other WPs/tasks

- T 3.2: Extract simple kernels from climate models
- T 3.1: (Survey and) selection of test cases

Proposed Strategy for 2016

Proposed approach: Incremental refinement of prototypes

- Build a first prototype with some features quickly
- Start the discussion with users having the prototype results
- Maintain a repository with source/target representations

2016

- M4: for each model some kernels and testcases are available
- M5: have a first design document for source-to-source tool
- M6: identified (first) user issues with code and wishlist
- M7: discussion between wishlist and tool development
- M10: prototype tool to translate DSLized code into original
- M12: Paper i.e. "Potential of DSLs for Icosahedral Models"

Discussion time...

Related Work for WP2

Related Work for WP2

Ongoing

- Analysis of lossless compression on Mistral
  - Compression ratio of 2:1 with LZMA
- API and library for lossy compression started
  - We expect a compression ratio of $>$ 10:1

Previous work

- Studies of compression ratio in e.g. ICOMEX

Analysis of Lossless Compression on DKRZ' Mistral

- Compression of LZMA, ZIP, SZIP, BZIP2 on production data
  - Selected a representative random sample
    - 35.4 TiB, 200k files of 13 PB, 280 M (0.27%, 0.75%)
  - Analyzed compression ratio

Relative size of the compressed file vs. files scanned

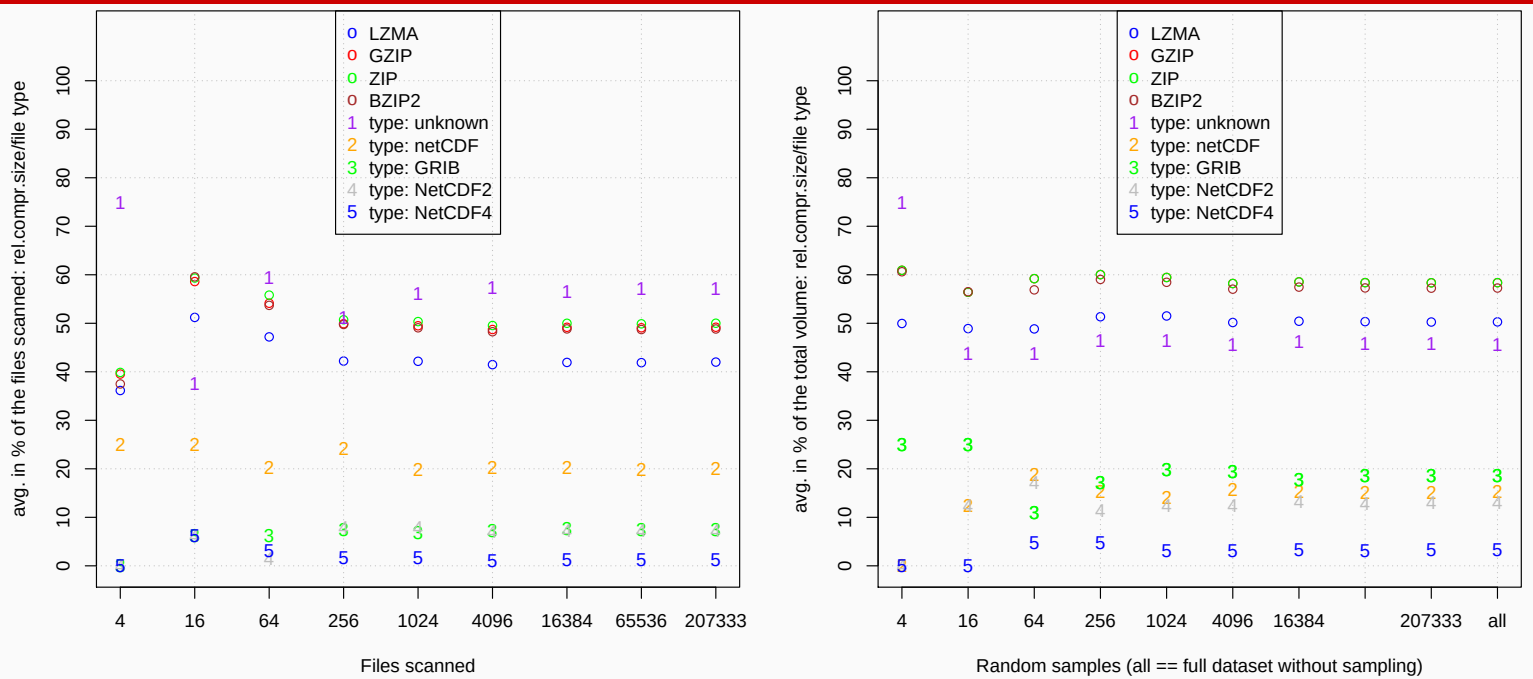


Figure : Left avg. across file number., right avg. over the data volume

Compression of Scientific Data Formats (LZMA)

- Mean compression ratio is 2.5:1 (0.5 rel. size)
- Questions:
  - What is the mean compression across all files?
  - What is the mean compression over the capacity/size?

Type	File #	Size in GiB	Compr. file	Compr. size
unknown	118366	16578.79	0.386	0.583
netCDF2	15117	4703.55	0.378	0.385
netCDF	41596	5567.41	0.414	0.412
netCDF4 classic	3081	258.51	0.838	0.789
netCDF4	2683	1204.19	0.801	0.482
GRIB	15453	6708.00	0.526	0.468
GRIB2	158	0.84	0.836	0.844
IEG	7463	504.14	0.615	0.559
SERVICE	1398	148.90	0.159	0.267
EXTRA	2018	593.80	0.319	0.377

SCIL: Scientific Compression Interface Library

- Ongoing work, started in an software lab (with Armin Schaare)
- C-Library for lossy compression of (dp) floating point
- User provides required precision
  - Relative error tolerance
  - Absolute error tolerance
  - Significant digits
- SCIL chooses the algorithm which meets these demands
- Likely, that we'll have performance constraints, too
- We can integrate such a library into e.g. HDF5

SCIL: Example Usage

```
scil_hints hints;
hints.absolute_tolerance = 1.0;
hints.significant_digits = 3;

scil_context * ctx;

ret = scil_create_compression_context(& ctx, & hints);
size_t comp_size;
char * comp_data = malloc(data_size +
    SCIL_BLOCK_HEADER_MAX_SIZE);
ret = scil_compress(ctx, comp_data, & comp_size,
    input_data, input_length);

// do sth. with comp_data

size_t orig_data_size;
char * orig_data = malloc(max_data_size);
ret = scil_decompress(orig_data, & orig_data_size,
    comp_data, comp_size);
```

Remeber: Lossy Compression Study in ICOMEX

Study of GRIB2, GRIB2 + JPEG2000, GRIB2 + LZMA, APAX

- APAX: Cooperation with SAMPLIFY (now out of business)

GRIB2 Compression Scheme

- Choose the number of bits (accuracy)
- Identify Min/Max of FP values
- Quantize values to $h = (max - min)/2^{bits}$
- Encode value v as integer i
  $\hat{v} = min + i \cdot h$

Analyzing 200+ Output Variables of ECHAM

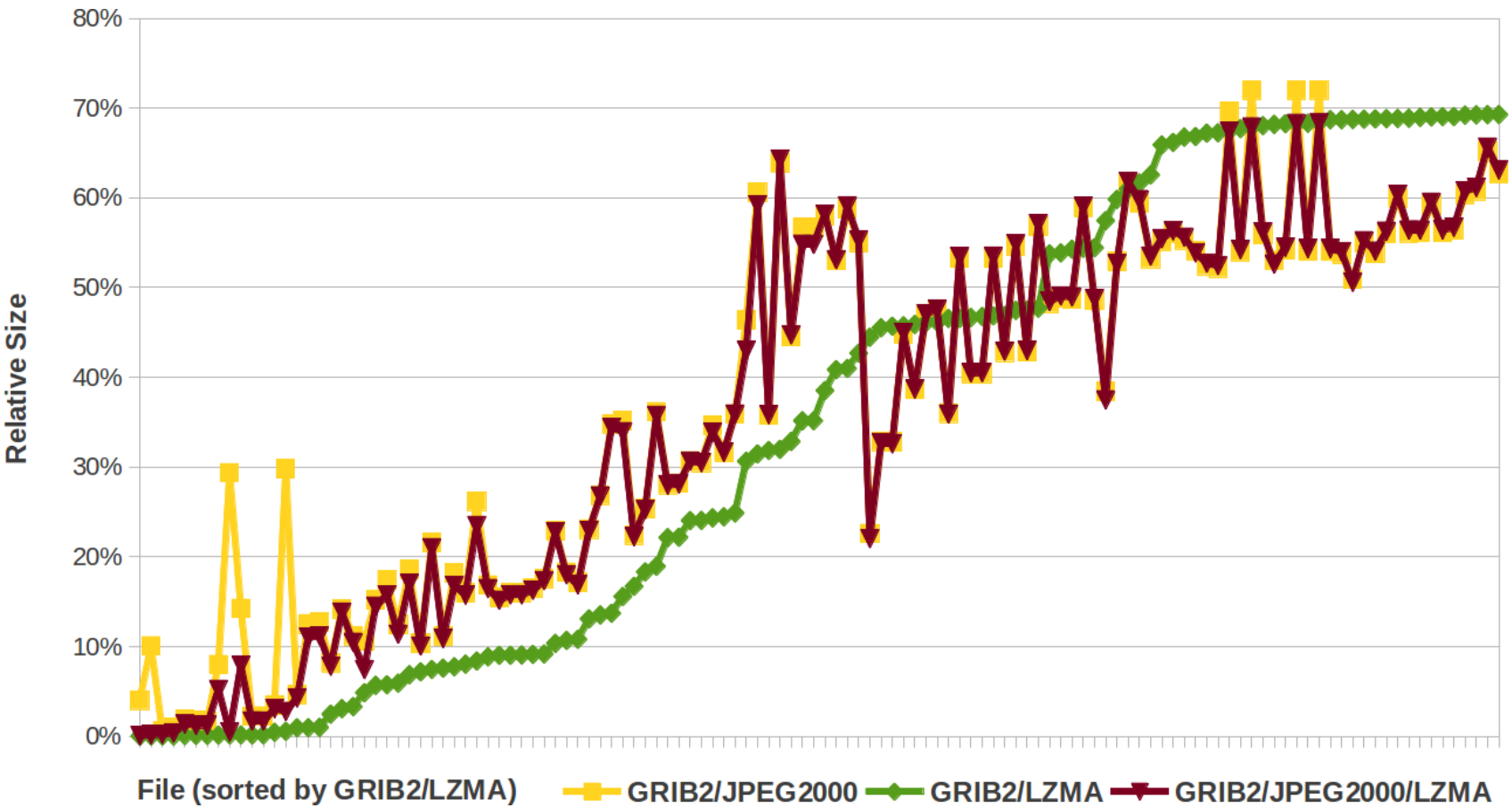


Figure : Relative compression size of variables sorted by GRIB2/LZMA

Consequence: One lossy compression scheme is not sufficient