Scientific Data Compression with SCIL

Julian M. Kunkel[1,2], Nabeeh Jum'ah[2], Anastasiia Novikova[2]

1: Department of Computer Science, University of Reading
2: Scientific Computing, Department of Informatics, University of Hamburg

2019-10-23

[3]Thanks to Thomas Dubos, Hisashi Yashiro for supporting this work.

Motivation

Large Data Volumes

- Weather and climate data is huge (30+ PByte for CMIP6)
  - Preserving the data is expensive (but necessary)
- Data compression can reduce the burden
  - But there exist many solutions for data compression
  - Users must hardcode the compression algorithm to use

Goals of AIMES:

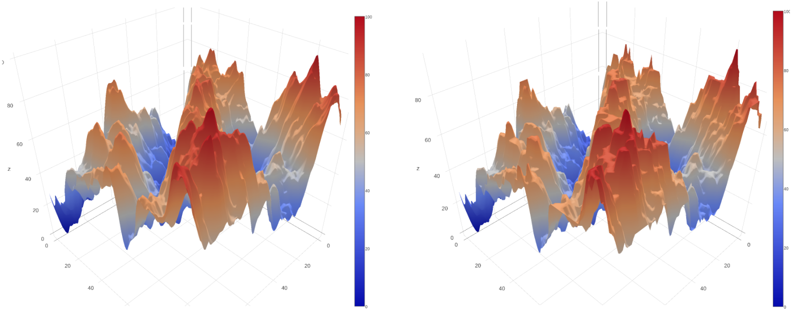
- Design of domain-specific compression (ratio $> 10 : 1$)
- Separation of concerns
  - User specifies required behavior, i.e., precision, performance
  - Compression library picks appropriate compression method

Approach

- Defined quantities to define accuracy/performance
- Design of the Scientific Compression Interface Library (SCIL)
  - Implementation of suitable compression schemes
  - Supporting existing compression algorithms
- Integration of SCIL into HDF5/NetCDF
- Evaluation of the performance on various data sets
  - Climate/Weather data including NICAM
- Develop a methodology for identifying the required accuracy
  - Explore the impact of compression on the scientific conclusions

Example for Data Compression

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



Right image uses lossy compression (Sigbits 3bits, ratio 11.3:1)

Outline

Supported Quantities

Accuracy quantities:

absolute tolerance: compressed can become true value $\pm$ absolute tolerance

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

relative error finest tolerance: value defining 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

Performance quantities:

compression speed: in MiB or GiB, or relative to network or storage speed

decompression speed: in MiB or GiB, or relative to network or storage speed

Supplementary quantities:

fill value: a value that scientists use to mark special data point

Overview of SCIL Features

- API for setting quantities
  - C, NetCDF, HDF5 compatibility
  - NetCDF variables can also be defined in an external file
- Metacompressor
  - Can utilize state-of-the-art lossy and lossless compressors
  - Brings some own compressors
  - Can chain different components of compressors
- Additional tools
  - Compression (NetCDF4, NetCDF3, CSV)
  - Benchmarking
  - Pattern creator to create arbitrary sized synthetic data
  - Noise generation/addition
  - (Plotting)

Configuration File for the Quantities

The file in the env. var. NETCDF_SCIL_HINTS_FILE is read by NetCDF and applied to the respective variables:

```
Variable1:
 relative_tolerance_percent=1

Variable2:        # Developer comment1
 relative_tolerance_percent=1
 relative_err_finest_abs_tolerance=1
 absolute_tolerance=1
 significant_digits=1
 significant_bits=1
 lossless_data_range_up_to=1
 lossless_data_range_from=1
 fill_value=4711
 comp_speed=0.5*MiB
 decomp_speed=1*NetworkSpeed
 force_compression_methods=abstol,lz4
```

Architecture of SCIL

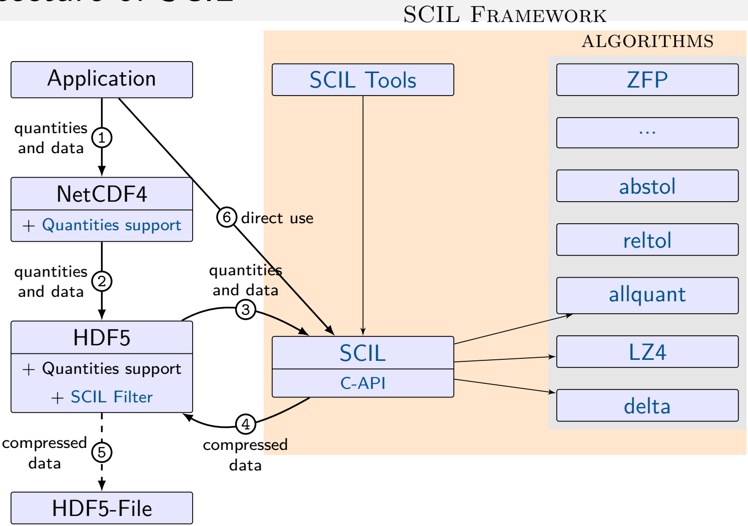
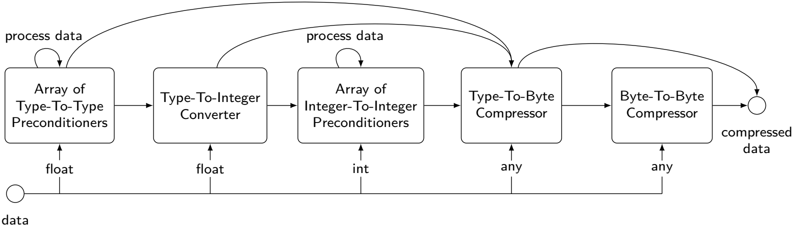


Figure: SCIL compression path and components (extended)

Compression Chain

- Chaining of compressors, e.g., preconditioner, entropy coder
  - Depends on the datatype: float, integer or bytes
  - Preconditoner: manipulate data to make it better compressible
  - Entropy coder: lossless compression scheme



Supported Compression Schemes

Existing algorithms

- Lossless: ZSTD, LZ4, GZIP, huffman
- Lossy: FPZIP, ZFP, SZ
  - They support only absolute or the relative precision quantity
- (Memcopy for benchmarking and to sustain performance)

Developed algorithms

- Abstol: Only absolute tolerance
- Reltol: Only relative tolerance
- Sigbits: Only significant bits
- Allquant: respect all precision quantities

Identifying Appropriate Algorithm: Applying the Quantities

- Significant digits and bits can be converted
  - Take the more precise one
- Relative tolerance can be converted to significant bits
  - Use absolute compression for finest tolerance (for small value)
- Treat fill value explicitly
- Min/Max are used by many methods to change behavior
- Check if compression would be fast enough

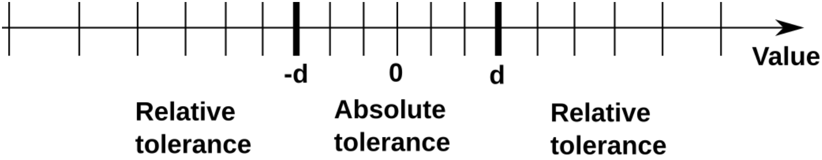


Figure: Range for Allquant, the sign is only stored if necessary

Outline

Analyzing Performance of Lossy Compression

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, Isabel, NICAM

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

Comparing Algorithms for the Scientific Files

	Algorithm	Ratio	Compr. MiB/s	Decomp. MiB/s
NICAM	abstol	0.206	499	683
	abstol,lz4	0.015	458	643
	sz	0.008	122	313
	zfp-abstol	0.129	302	503
ECHAM	abstol	0.190	260	456
	abstol,lz4	0.062	196	400
	sz	0.078	81	169
	zfp-abstol	0.239	185	301
Isabel	abstol	0.190	352	403
	abstol,lz4	0.029	279	356
	sz	0.016	70	187
	zfp-abstol	0.039	239	428
Random	abstol	0.190	365	382
	abstol,lz4	0.194	356	382
	sz	0.242	54	125
	zfp-abstol	0.355	145	241

(a) 1% absolute tolerance

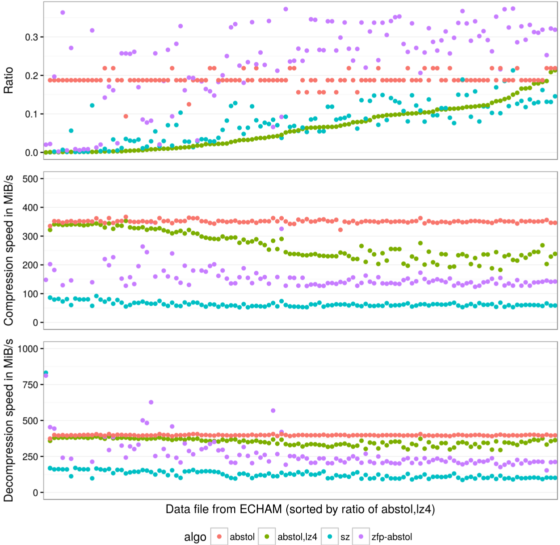
	Algorithm	Ratio	Compr. MiB/s	Decomp. MiB/s
NICAM	sigbits	0.439	257	414
	sigbits,lz4	0.216	182	341
	zfp-precision	0.302	126	182
ECHAM	sigbits	0.448	462	615
	sigbits,lz4	0.228	227	479
	zfp-precision	0.299	155	252
Isabel	sigbits	0.467	301	506
	sigbits,lz4	0.329	197	366
	zfp-precision	0.202	133	281
Random	sigbits	0.346	358	511
	sigbits,lz4	0.348	346	459
	zfp-precision	0.252	151	251

(b) 9 bits precision

Table: Harmonic mean compression performance for different scientific data sets

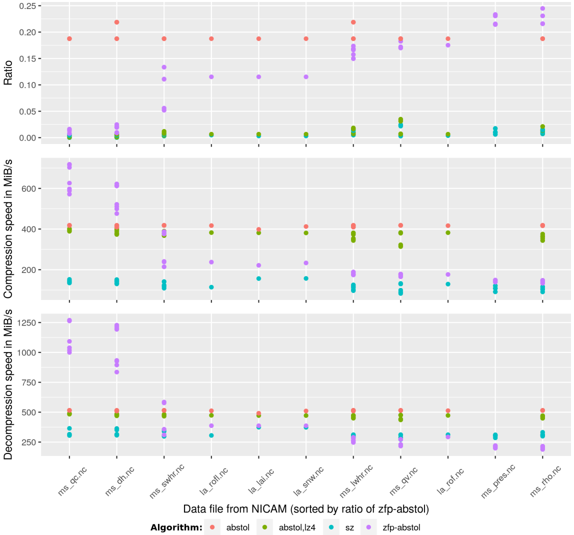
Results for Absolute Tolerance of ECHAM

- Using NetCDF3
- Abstol: 1% max
- Compress whole file
- Sometimes SZ better, sometimes abstol+lz4



Results for Absolute Tolerance of NICAM

- Using NetCDF4/HDF5
- Abstol: 1% max
- Multiple chunks generated
- Data is smooth:
  SZ best ratio
  Abstol+LZ4 fastest



Compression in HDF5 and NetCDF

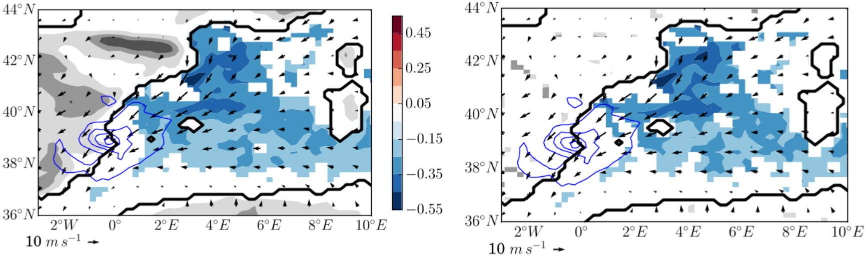
- Testing parallel compression on NetCDF capable DSL code
  - 128 processes on Mistral
- 268 million grid cells x 64 vercial levels
- Field values between -10 and +55

Compression method	Parameter	Write time	Data size	Throughput*	Speedup
SCIL		in s	in GB	in MB/s	
No-compression		165.3	71.4	432	1.0
memcopy		570.1	71.4	125	0.3
lz4		795.3	71.9	90	0.2
abstol,lz4	absolute_tolerance=1	12.8	2.7	5578	12.9
	absolute_tolerance=.1	72.6	2.8	983	2.3
sigbits,lz4	relative_tolerance_percent=1	12.9	2.9	5535	12.8
	relative_tolerance_percent=.1	18.3	3.2	3902	9.0

Table: Results. *Virtual throughput relative to the uncompressed file size.

- Access to shared file is slow, overhead of HDF5 compression chain
- Parallel compression lately added to HDF5

Investigating Compression Error



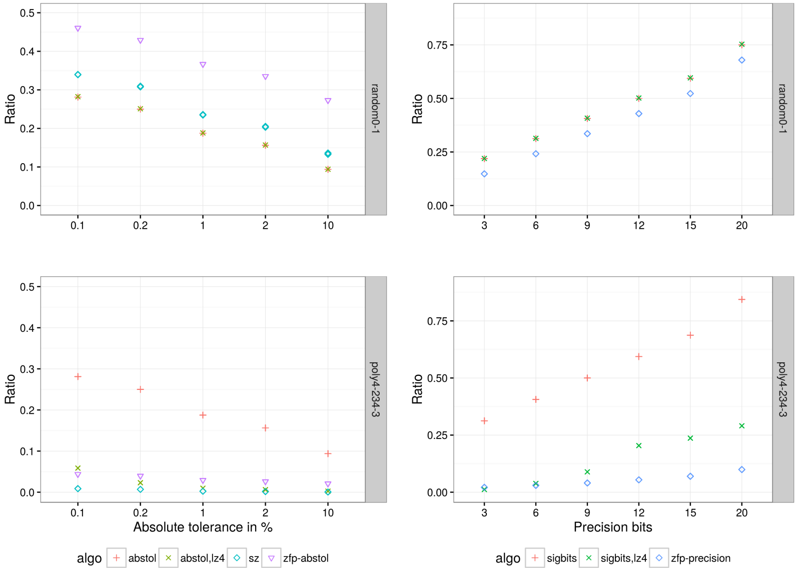
- Blue shade: conditional mean SST difference between CPL and SMO simulations where statistical test is passed
- Adding noise affects little the overall pattern, but makes it harder to pass statistical test (more white holes in the blue)
- Conclusions unchanged even with quite large noise on SST (0.2°K), probably due to averaging several events

Outline

Summary

- SCIL addresses compression (lossy and lossless)
  - Allow scientists to define required variable accuracy
  - Exploit this knowledge in the compression scheme
  - Novel schemes sometimes compete with existing algorithms
- Predicting performance/ratio for data is difficult
- Future work: machine learning of
  - Performance, expected ratio for data
  - Best compression algorithms

Synthetic Patterns



Investigate Tolerable Error

- Under noise mimicking compression, reproduce conclusions of Berthou et al. 2016[1]
  - Mediterranean region (Spain, France)
  - Evaluates how wind, through its action on the ocean, impacts (with some delay) heavy precipitation
  - Compares outputs from two simulations (CPL and SMO) and subjects the difference to tests of statistical significance

    Student t test; 97.5% probability of rejecting a zero difference
  - Analyzed fields: wind, rain (convective and non-convective), humidity, sea-surface temperature (SST)

[1] DOI 10.1007/s00382-016-3153-0

Approach

1. Reproduce published results (Find data/scripts used in paper)
2. Apply statistical model for noise induced by lossy compression
   - Use Gaussian white noise
3. Redo the analysis, check if the conclusions are still supported
4. Increase levels of noise to input data (= model output)
   - One field at a time; then together