IN53B-0827 Improving Scaling Properties of Common Statistical Operators for Gridded Geoscience Datasets

Charlie Zender <zender@uci.edu> and Harry Mangalam, Department of Earth System Science, University of California, Irvine
Daniel Wang <wangd@uci.edu>, Department of Electrical Engineering and Computer Science, University of California, Irvine

Abstract:

An accurate cost-model that accounts for dataset size and structure can help optimize geoscience data analysis. We develop and apply a computational model to estimate data analysis costs for arithmetic operations on gridded datasets typical of satellite- or climate model-origin. For these dataset geometries our model predicts data reduction scalings that agree with measurements of widely-used geoscience data processing software, the netCDF Operators (NCO) (*Zender*, 2006). I/O performance and library design dominate throughput for simple analysis (e.g., dataset differencing). Dataset structure can reduce analysis throughput ten-fold relative to same-sized unstructured datasets. We demonstrate algorithmic optimizations which substantially increase throughput for more complex, arithmetic-dominated analysis such as weighted-averaging of multi-dimensional data. We show how these principles accelerate terascale data reduction by benchmarking the time for NCO to characterize the variability of climate simulations from the Intergovernmental Panel on Climate Change (IPCC) fourth assessment report (AR4).

Motivation:

Model intercomparisons, such as the IPCC AR4 climate assessment, are too cumbersome for individual researchers to easily identify and explore interesting characteristics. Data storage and analysis strategies use cost models identify bottlenecks (e.g., file transfers, arithmetic) in advance and optimizing that processing stage (e.g., asynchronous pre-fetching, retention and re-use of intermediate results).

Strategy:

Explicitly compute data reduction time T of common statisitical operations as functions of integer I, floating point F, and I/O operations and their speeds v_i:

$$T = T_{\rm I} + T_{\rm F} + T_{\rm W} + T_{\rm R} \quad (1a)$$
$$= I/v_{\rm I} + F/v_{\rm F} + N_{\rm W}W/v_{\rm W} + N_{\rm R}W/v_{\rm R} \quad (1b)$$

Computational Model Results:

Binary operations (e.g., differencing datasets) are I/O-dominated. I and F depend only on dataset size N:

$$F(\text{binary}) = N \quad (2a)$$
$$I(\text{binary}) \approx 3N(W+2) \quad (2b)$$

Weighted averages are arithmetic-dominated and depend strongly on rank R:

$$F = N(3 + 2N_{\rm A}^{-1}) \quad (3a)$$
$$I_1 \approx N[34R + 8R_{\rm w} + 25 + W + (W+11)N_{\rm A}^{-1}] + B \quad (3b)$$
$$I_2 \approx N[28R + 0 + 23 + W + (W+11)N_{\rm A}^{-1}] + B \quad (3c)$$
$$I_3 \approx N[\ 6R + 8R_{\rm w} + 17 + W + (W+11)N_{\rm A}^{-1}] + B \quad (3d)$$
$$I_4 \approx N[\ \ 0 + 0 + 15 + W + (W+11)N_{\rm A}^{-1}] + B \quad (3e)$$

where I_1 is the operation count required without optimizations, and I_2, I_3, and I_4, include the WRU, MRV, and both optimizations, respectively. Here $R_{\rm w}$ is the rank of the weight (e.g., gridcell area), W is the data wordsize (i.e., four or eight-bytes), and $N_{\rm A}$ and $N_{\rm w}$ are the products of the sizes of the averaged and weight dimensions, and $B = (W+2)N_{\rm w}$.

Table 1: **Operation Count Notation**

Parameter	Description
F	Floating point operations
I	Total integer operations
N	Number of input data
$N_{\rm A}$	Input data per output datum
R_i	Rank of variable
T_i	Operation time [s]
v_i	Operation speed [# s^{-1}]
W	Datum size [B]

Benchmark Results:

Compare measured operation counts to model (2)–(3) for differencing and weighted-averaging prototype Satellite and IPCC GCM datasets (Table 2):

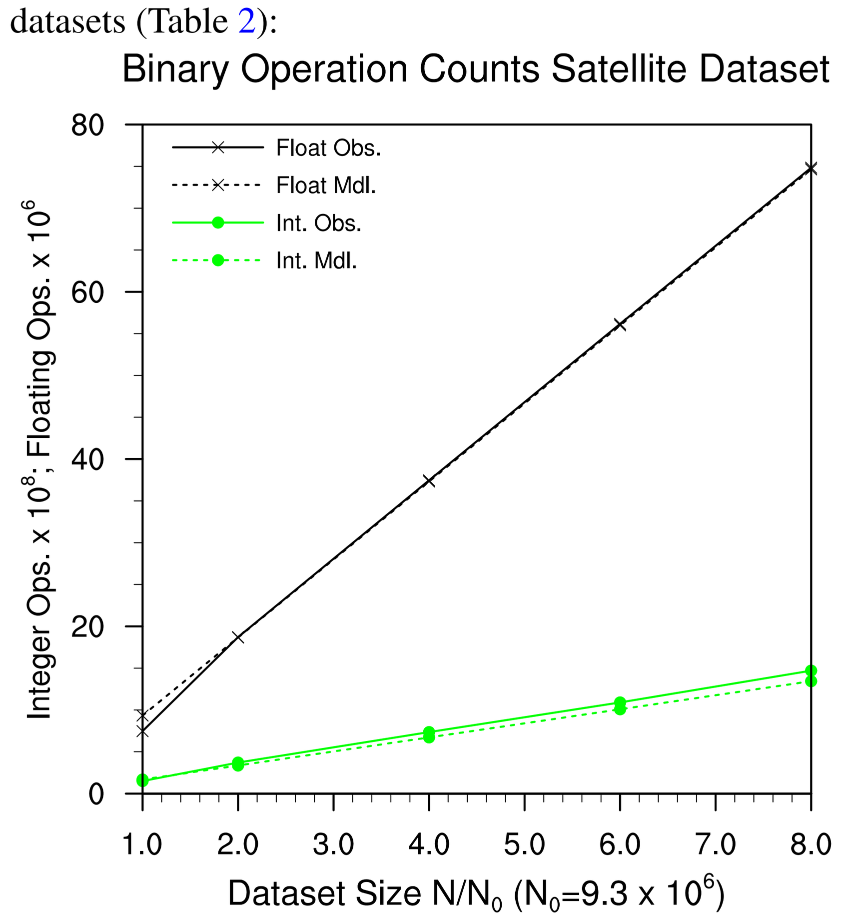

Figure 1: Observed (solid) and predicted (dashed) integer I and floating point F operations necessary to difference datasets with N elements. Horizontal axis scaled to units of $N_0 = 9.3 \times 10^6$ elements. Note different scales for integer and floating point operations.

- Binary operations (2) easy to predict (and hard to optimize!)

Weighted averaging requires memory broadcasting broadcasting and collection that are expensive, though amenable to optimization for certain computational geometries:

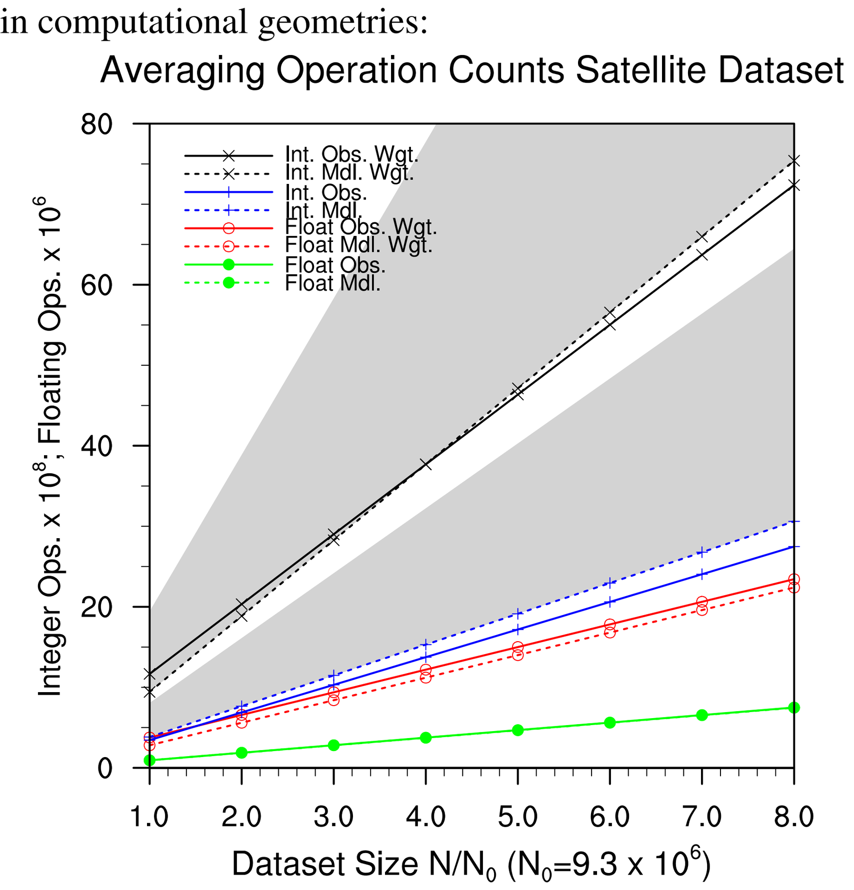

Figure 2: Observed (solid) and predicted (dashed) integer I and floating point F operations necessary to average an N element dataset with (upper two sets of curves) and without (lower two sets of curves) weighting. Grey areas indicate I predicted for rank R = 2–5 datasets. Other markings as in Figure 1.

- Integer operations dominate floating point
- Dataset structure (i.e., rank R) drives cost

Prototype Datasets to Benchmark:

Typical NASA satellite and IPCC GCM dataset geometries:

Table 2: **File Geometries**

	Satellite	GCM
Max. Rank R	2	4
Variables	8[a]	128[b]
Time	—	8
Level	—	32
Latitude	2160	128
Longitude	4320	256
Elements N [#]	75×10^6	285×10^6
Total Size [MB]	299	1143

[a] All eight variables are rank $R = 2$.
[b] Eight variables are scalars ($R = 0$), eight variables contain the time dimension only ($R = 1$), sixteen variables contain only latitude and longitude ($R = 2$), sixty-four variables have time, latitude, and longitude ($R = 3$), thirty-two variables have contain all four dimensions ($R = 4$).

Throughput Including Arithmetic and I/O:

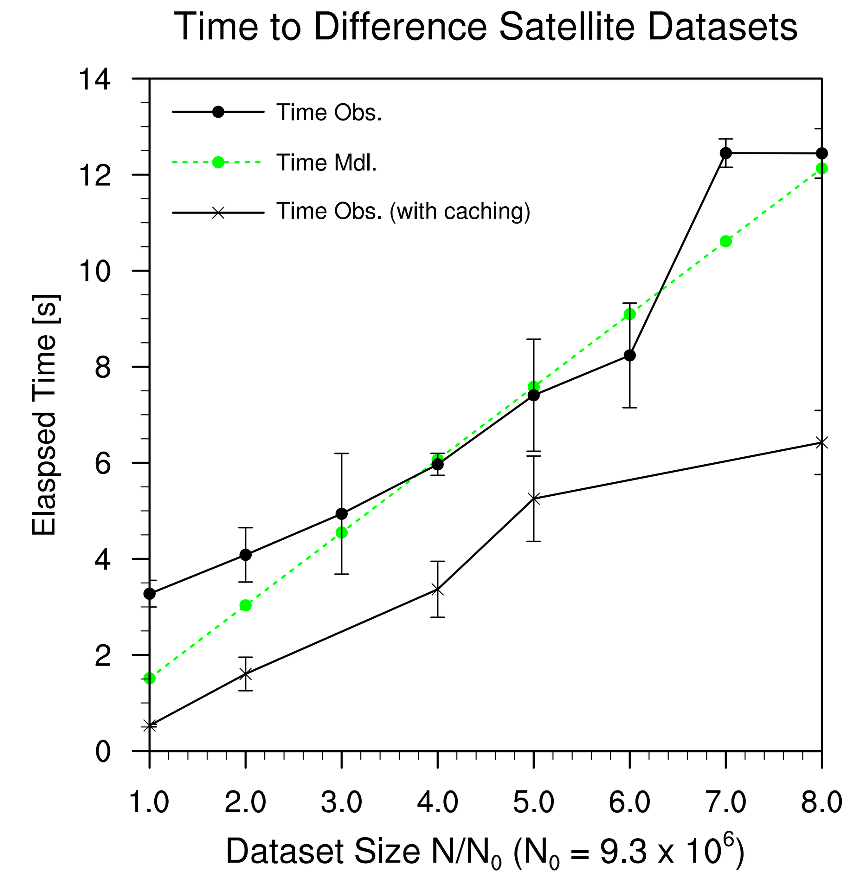

Figure 3: Elapsed time T [s] to difference two datasets each with N elements. Measurements (solid) are for non-cached and cached datasets, model (dashed) is for non-cached only. Vertical bars span range of four replicate measurements. Other markings as in Figure 1.

- Disk-level data-caching helps significantly (probably more in benchmarks than real world)

The average $\bar{x}$ of a variable x weighted by w with mask m and missing value μ is

$$\bar{x}_j = \sum_{i=1}^{i=N_{\rm A}} \mu_i m_i w_i x_i \Bigg/ \sum_{i=1}^{i=N_{\rm A}} \mu_i m_i w_i \quad (4)$$

where the subscripts i and j range over the input and averaged hyperslabs, respectively. What does weighting cost?

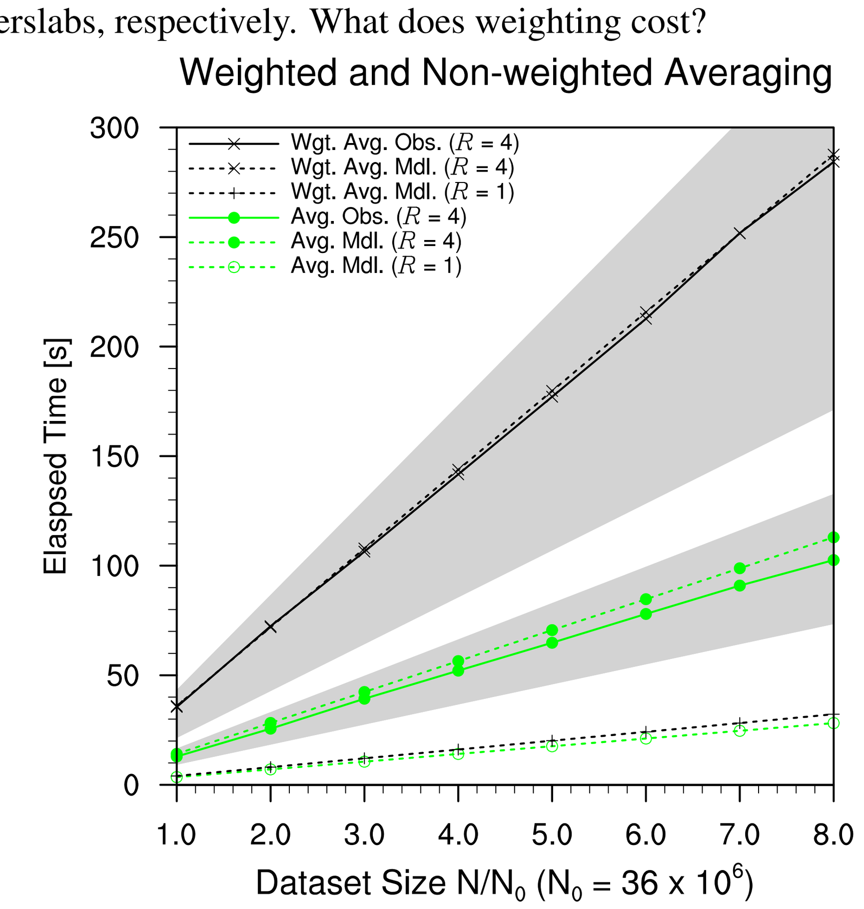

Figure 4: Observed (solid) and predicted (dashed) elapsed time to perform weighted and non-weighted averages on N element GCM-geometry ($R = 4$) and unstructured ($R = 1$) datasets. Grey areas indicate prediction range for rank R = 2–5 datasets. Horizontal axis scaled to units of $N_0 = 36 \times 10^6$ elements. Other markings as in Figure 1.

- Computational model (3b) accurately predicts throughput for all dataset ranks and sizes
- Weighted averages take about about three times longer than un-weighted for rank $R = 4$ datasets

Optimization:

Two powerful optimizations for weighted rank reduction are:

1. Weight Re-Use (WRU): Re-use broadcast weights to average other variables of the same shape
2. Most-Rapidly-Varying (MRV) optimization: Eliminate cost of collecting averaging hyperslabs when averaging order equals storage order

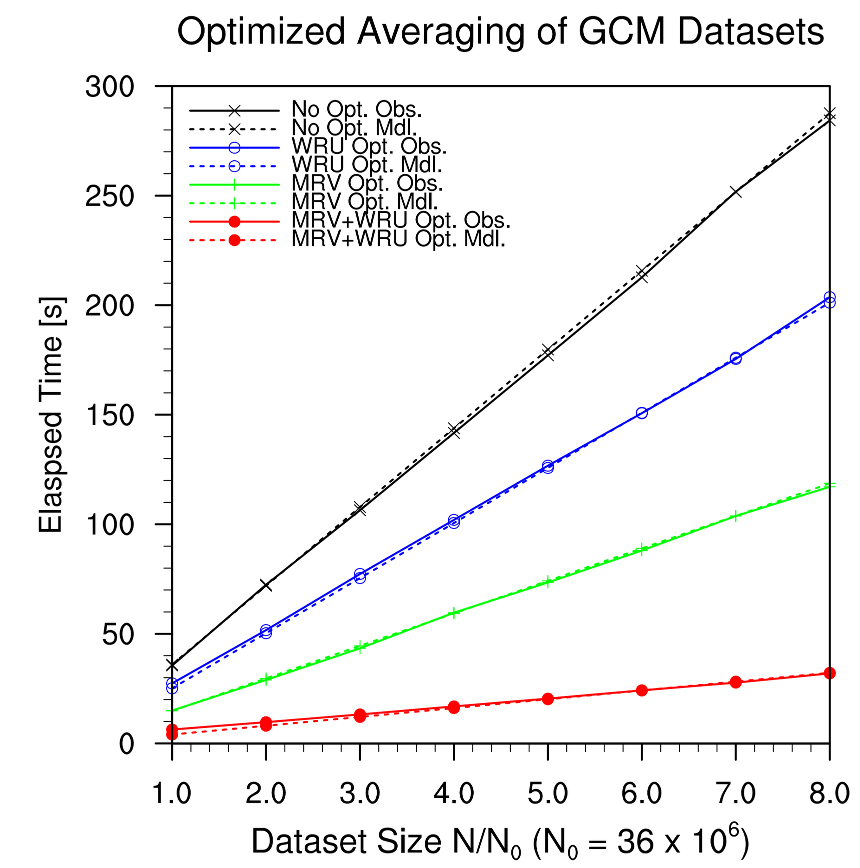

Figure 5: Observed (solid) and predicted (dashed) elapsed times to perform a weighted average of an N element GCM-geometry ($R = 4$) dataset with and without WRU and MRV and both optimizations. Un-optimized curves same as Figure 4. Other markings as in Figure 1.

- WRU reduces black line (3b) to blue line (3c) in Figure 5
- MRV reduces black line (3b) to green line (3d)
- Combining WRU and MRV reduces black line (3b) to red line (3e)

Application: IPCC AR4 Model Intercomparison

We applied NCO to intercompare 21st century climate predictions from 16 GCMs, a prototypical problem in terascale data reduction.

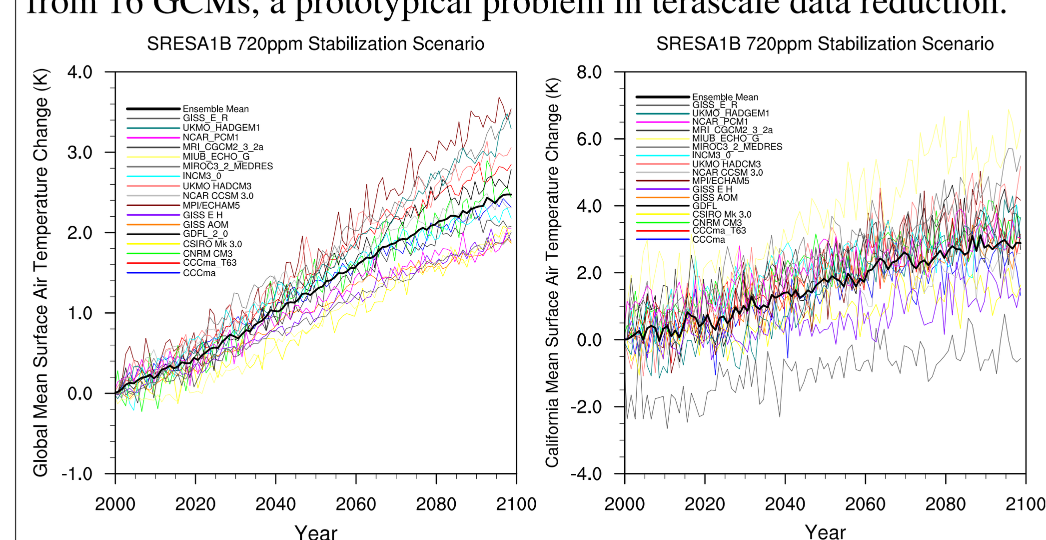

Figure 6: Predicted Global (left) and California (right) annual-mean temperature from 2000–2099 under SRESA1B 720 ppm CO_2 stabilization scenario. Temperature scales differ.

- Averaging steps 5–10× faster as described above
- Script Workflow Analysis for MultiProcessing (SWAMP, see Poster IN53B-0826) accelerates entire script by smart scheduling based on dependency tree analysis
- Earth is quickly warming—must reduce data faster!

Conclusions:

Our algorithms accelerate unoptimized data reduction about tenfold. This improvement is generic in that the same algorithms and operators apply to datasets from any geoscience model producing gridded, multi-dimensional datasets of similar rank. The computational model agrees with throughput measurements for "building block" arithmetic operations which compose most complex statistical operations such as standard deviations, t-tests, etc. We are one step closer to intelligent scheduling (local vs. server-side) of arbitrarily complex data reduction in distributed environments.

Future Directions:

- Workflow Analysis and Optimization with SWAMP (Script Workflow Analysis for MultiProcessing, see Poster IN53B-0826)
- Parallel I/O with pnetCDF and netCDF4
- Parallelize basic blocks with dependency-tree analysis
- Server-Side Data Reduction and Analysis

Operator Shared and Distributed Parallelism:

All arithmetic operators support Shared Memory Parallelism (SMP) and distributed parallelism:

Table 3: **Operator Parallelism Summary**

Command	Functionality	Type[a]	MFO[b]	Par.[c]
`ncap`	Arithmetic Processor	A		
`ncatted`	Attribute Editor	M		
`ncbo`	Binary Operator	A		✓
`ncea`	Ensemble Averager	A	✓	✓
`ncecat`	Ensemble Concatenator	M	✓	✓
`ncflint`	File Interpolator	A		✓
`ncks`	Kitchen Sink	M		
`ncpdq`	Pack and Permute Data	A/M		✓
`ncra`	Record Averager	A	✓	✓
`ncrcat`	Record Concatenator	M	✓	✓
`ncrename`	Renamer	M		
`ncwa`	Weighted Averager	A		✓

[a] Operator type: "A" and "M" indicate arithmetic and metadata operators, respectively.
[b] Multi-file Operators—Operators which process an arbitrarily large number ($N > 2$) of input files.
[c] Operator parallelism. These operators exploit shared memory parallelism (SMP) on OpenMP-compliant platforms, and distributed parallelism with MPI.

- MPI and OpenMP parallelism for arithmetic is effective
- netCDF serial write requirements prevents parallel writes
- May surmount I/O bottleneck with pnetCDF (*Li et al.*, 2003) or netCDF4 (*Rew et al.*, 2006)

Acknowledgements:

Thanks to Unidata for creating and maintaining netCDF. Supported by NSF Grants ATM-0231380 and IIS-0431203. This poster is available for download at:
`http://dust.ess.uci.edu/ppr/pst_ZMW06.pdf`

References

Li, J., et al. (2003), Parallel netCDF: A high-performance scientific I/O interface, in *Proceedings of the 2003 ACM/IEEE Conference on Supercomputing*, pp. 39–49, Association for Computing Machinery, IEEE Computer Society, Washington, DC, USA, November 15–21, Phoenix, AZ.

Rew, R., E. Hartnett, and J. Caron (2006), NetCDF-4: Software implementing an enhanced data model for the geosciences, in *Proceedings of the 22nd AMS Conference on Interactive Information and Processing Systems for Meteorology*, p. 6.6, American Meteorological Society, AMS Press, Boston, MA.

Zender, C. S. (2006), netCDF Operators (NCO) for analysis of self-describing gridded geoscience data, *Submitted to Environ. Modell. Softw.*, available from `http://dust.ess.uci.edu/ppr/ppr_Zen07.pdf`.

Zender, C. S., and H. J. Mangalam (2007), Scaling properties of common statistical operators for gridded datasets, *In Press in Int. J. High Perform. Comput. Appl.*, available from `http://dust.ess.uci.edu/ppr/ppr_ZeM07.pdf`.