How Much can we Really Compress Scientific Data?

Franck Cappello

Argonne National Laboratory

University of Illinois at Urbana Champaign

Arkaprabha Ganguli (Michigan State University),
David Krasowska(Clemson),
Julie Bessac (ANL), Sheng Di (ANL), Robert Underwood (ANL),
Xiaodong Yu (ANL,
Jon C. Calhoun (Clemson)

Progress

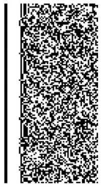
CCDSC 2016

Argonne NATIONAL LABORATORY

Scientific Data Compression: From Stone-Age to Renaissanc

- Background
- Focus on spatial cor
- Best in class lossy c
- Open questions

This is what we need to compress (bit map of 128 floating point numbers):



CCDSC 2018

Three Frontiers of Lossy Compression of Scientific Data

Three Frontiers

Past (still used today)

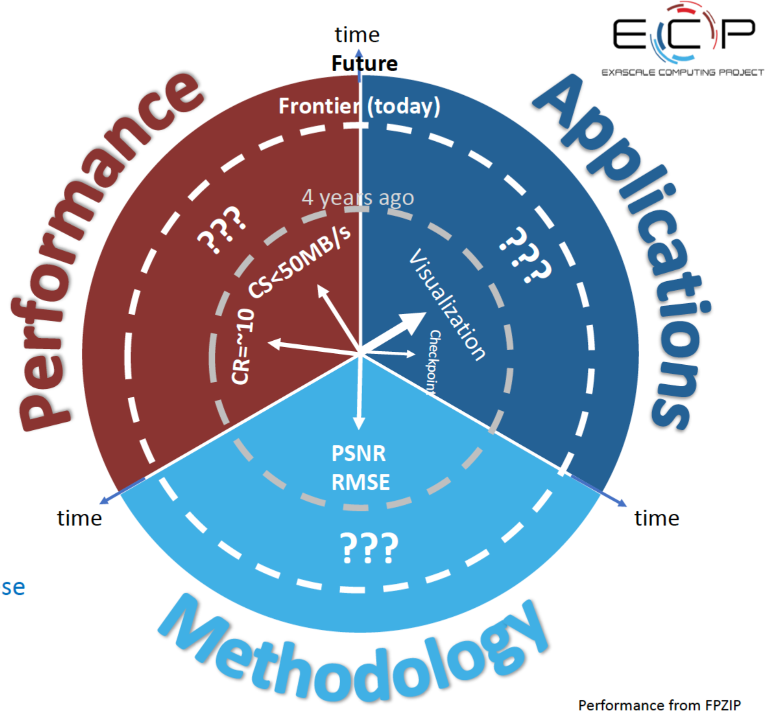
-Lossless compression

-Decimation in space and time + linear or tricubic (or other) interpolation

-Lossy compression mainly for Visualization

Technologies that are at the current frontiers: SZ, ZFP, Z-checker+SDRBench

In this talk, I will illustrate the progresses in these three frontiers based on our explorations and results in the ECP EZ, CODAR and Exasky Projects and ANL SZ compressor



Performance from FPZIP Miranda dataset

Argonne NATIONAL LABORATORY

Lossy compression of scientific data

- Consist in reducing scientific data volume by leveraging correlations and reducing precision (lossless compression does not reduce scientific data enough)

- Compression ratios (with current compressors) vary depending on use-cases, typically:
  - CR=5 for hard to compress dataset and demanding data/analysis quality preservation
  - CR=10-100 for scientific data presenting high correlation and medium data/analysis quality preservation
  - CR=x100 for visualization (low data/analysis quality preservation)

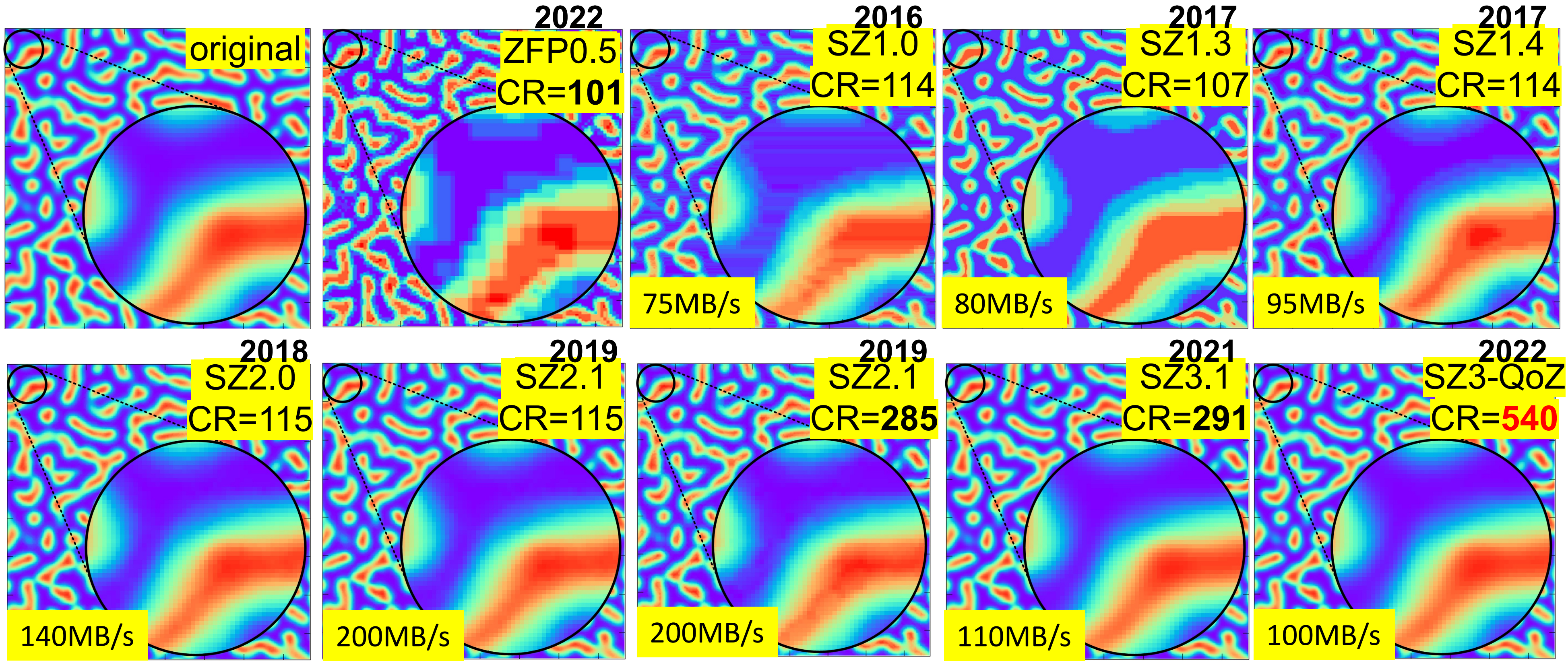
- Goal: keep the same science (satisfy user's quality requirements WRT QoIs – features)
  - WARNING: You will see images because this is the easiest way to show distortion but compression of scientific data is NOT for images

- Getting significant traction in the scientific community (climate, cosmology, seismic, etc.), IoT community as well (sensors, EKG)

Progress in Compression Techniques

Huge Progress in performance in the past 5-6 years

Evolution of SZ compression quality and performance using a large-eddy simulation of multicomponent flows with turbulent mixing: Miranda - density field.



Visualization of Miranda - density data for SZ's different versions (EB: VRAE 1E-2), Performance on single core CPU (Intel Broadwell)

SZx compresses at 300GB/s on NVIDIA A100 → Bottleneck is not compression but PCIe

More Lossy Compressors

ZFP (LLNL): Transform (DCT)
ECP ZFP
Overpreserves data, lower Compression ratio compared to SZ, Better speed.

SPERR (NCAR): Wavelet
Works well on wave propagation problem (Climate, Seismic)

MGARD (ORNL)
ECP CODAR
Multigrid adaptive reduction
MGARD controls the compression errors in quantities of interest (Q): Linear expression of the error

Largest Compression Ratio For Each Compressor that Satisfies Each Pinard et al (2020) Requirements

Compressor	Pearson R^2	Spatial Error	KS-test
SZ_Interp	93	93	**21**
SZ (regression)	14.34	14.34	**14.34**
ZFP	5.45	5.45	**2.36**
MGARD	27.1	4.69	X
MGARDx	14.7	6.49	X
TThresh	16.1	16.1	**2.98**
BitGrooming	1.51	1.51	**1.51**
Digit Rounding	1.86	1.86	**1.86**
FPZip	1.95	1.95	**1.95**
NDZip	1.64	1.64	**1.64**
Zstd	1.35	1.35	**1.35**

More Lossy Compressors

TTRESH (LLNL):

HoSVD (Tucker Decomposition)
Quantize the Core tensor
Very high compression ratio
Tendency to blur the overall data (loose details)
1 or 2 orders of magnitude slower than SZ or ZFP

Autoencoders

Overall architecture of convolutional autoencoder (A. Glaws, R. King, and M. Sprague, "Deep learning for in situ data compression of large turbulent flow simulations," Physical Review Fluids, vol. 5, no. 11, p. 114602, 2020.)

12 residual blocks for feature extraction + 3 compression layers

Largest Compression Ratio For Each Compressor that Satisfies Each Pinard et al (2020) Requirements

Compressor	Pearson R^2	Spatial Error	KS-test
SZ_Interp	93	93	**21**
SZ (regression)	14.34	14.34	**14.34**
ZFP	5.45	5.45	**2.36**
MGARD	27.1	4.69	X
MGARDx	14.7	6.49	X
TThresh	16.1	16.1	**2.98**
BitGrooming	1.51	1.51	**1.51**
Digit Rounding	1.86	1.86	**1.86**
FPZip	1.95	1.95	**1.95**
NDZip	1.64	1.64	**1.64**
Zstd	1.35	1.35	**1.35**

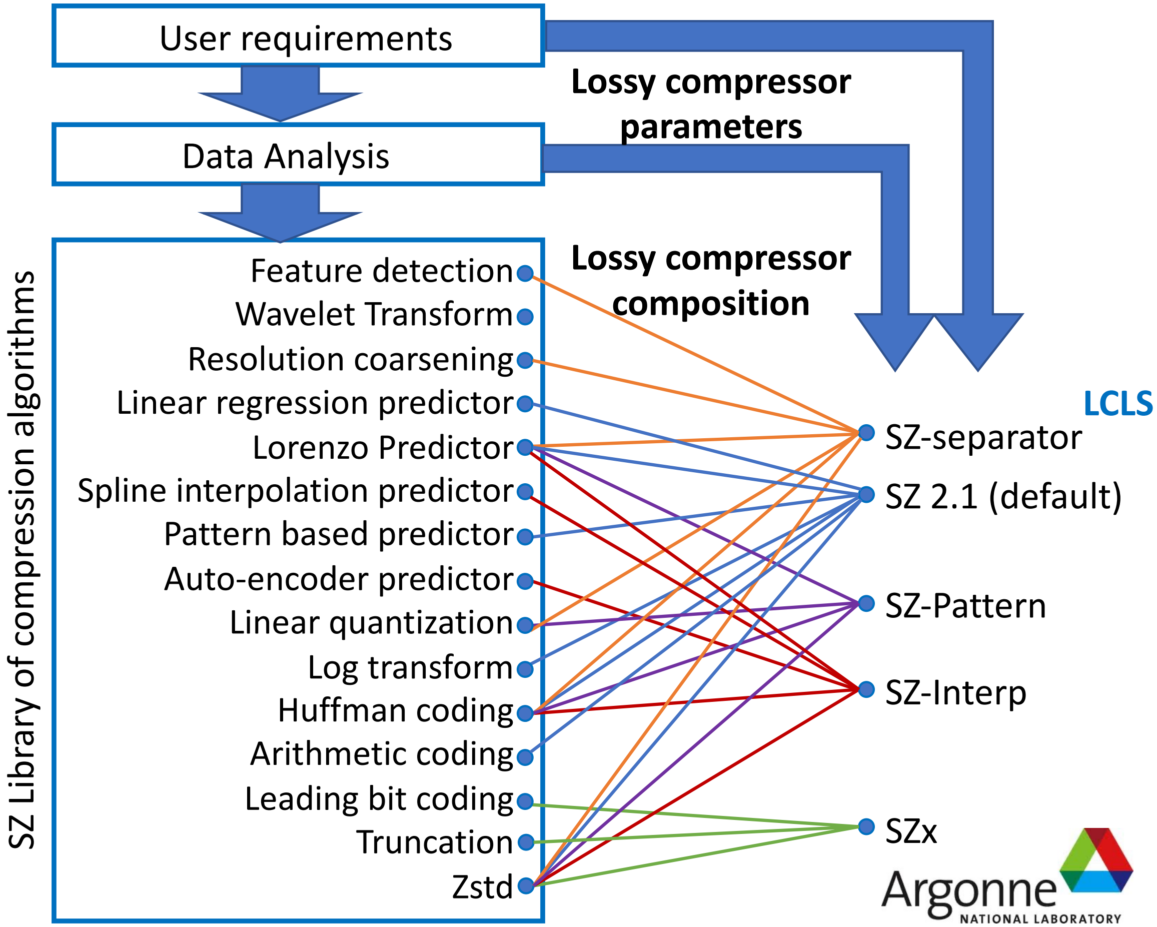
Significant Smoothing

What makes SZ3 different: a Highly **Modular/ Customizable** Compression Framework

2021 R&D 100 WINNER

SZ 3 (C++) library of algorithms for lossy compression and examples of SZ compressors built from the library of algorithms.

To compose and tune a compression pipeline we analyze the data to compress and user requirements in compression speed, ratio and accuracy.



Progress in Applications and Methodology

Many more Applications (than in 2018)

- Climate
- Combustion
- Cosmology
- Deep Learning
  - Activation data
  - Model coefficients
  - Training data
- Extreme Weather
- Fusion Energy
- Hydrodynamics
- IoT
- Light Sources (Physics Instruments)
- Materials Science
- Molecular Dynamics
- Quantum Chemistry
- Quantum Circuit Simulation
- Seismic Imaging

Many more Use-cases (than in 2018)

We are seeing an increasing diversity/number of use-cases

“Classic” use-cases:

1) Visualization

2) Reducing storage footprint (offline compression)

3) Reducing I/O time (on-line, in-situ compression)

Recently identified use-cases:

4) **Reducing streaming intensity** (recent for generic floating-point compressors)

5) Lossy checkpoint/restart from lossy state

- reduce checkpoints footprint on storage – adjoint, accelerate checkpointing

6) Re-computation Avoiding by reducing the memory footprint → GAMESS

7) **Running larger simulations by reducing the memory footprint**

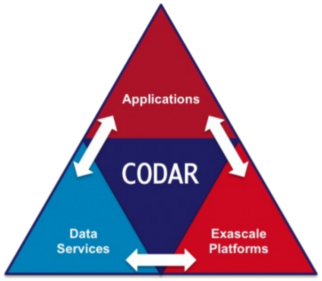
8) Accelerating CPU/GPU – memory transfer

9) Reduce DNN model size

10) Accelerate training (I/O read time) of DNNs

Cappello, F., Di, S., Li, S., Liang, X., Gok, A. M., Tao, et Al., Use cases of lossy compression for floating-point data in scientific data sets. *The International Journal of High Performance Computing Applications*, *33*(6), 1201–1220, 2019

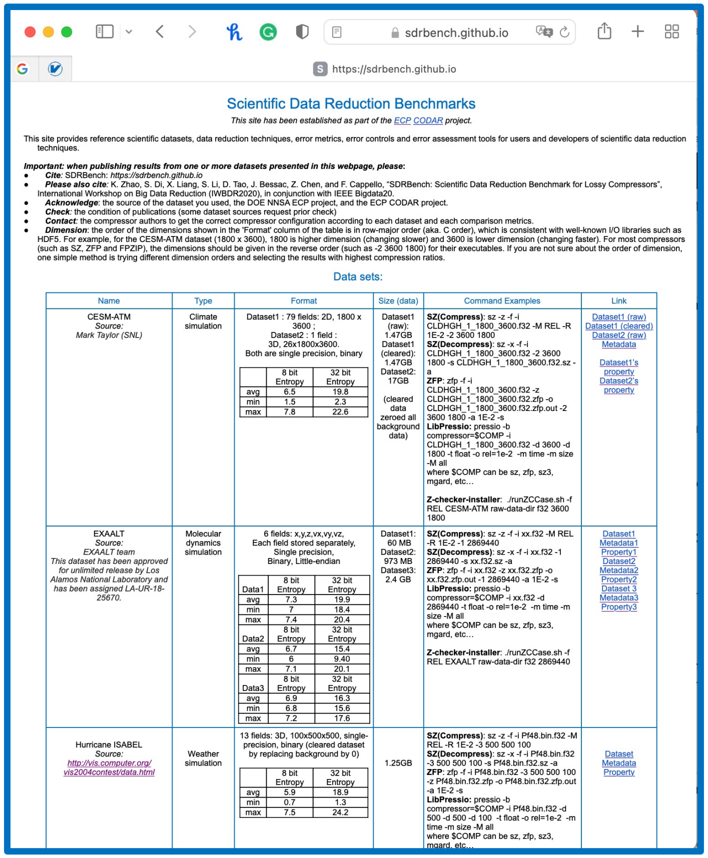
Huge Progress in Methodologies



ECP
EXASCALE COMPUTING PROJECT

https://sdrbench.github.io/

https://github.com/robertu94/libpressio



Scientific Data Reduction Benchmarks

This site has been established as part of the ECP CODAR project.

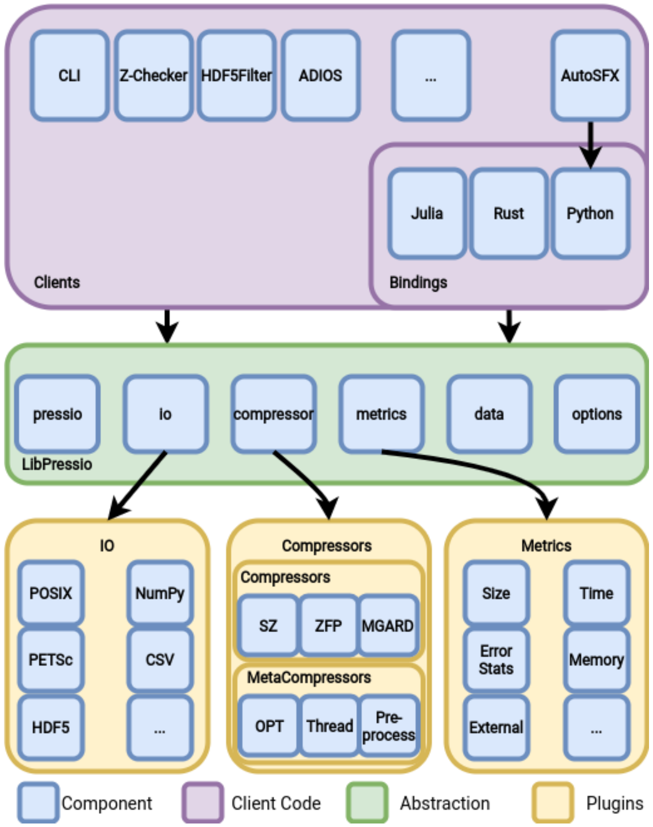
This site provides reference scientific datasets, data reduction techniques, error metrics, error controls and error assessment tools for users and developers of scientific data reduction techniques.

***Important: when publishing results from one or more datasets presented in this webpage, please*:**

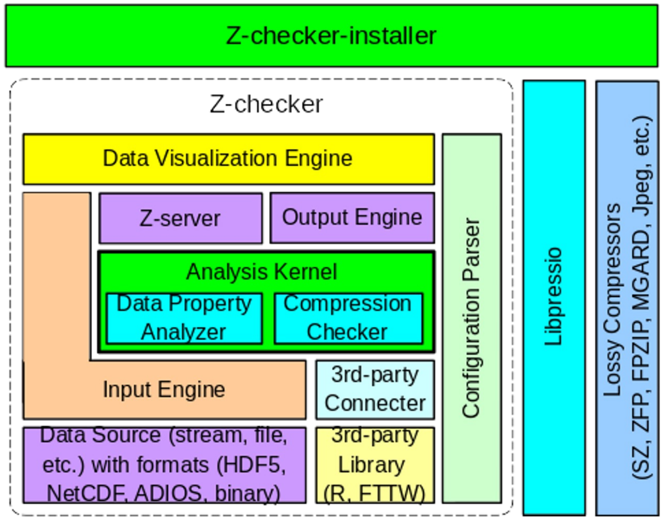
- ***Cite***: SDRBench: *https://sdrbench.github.io*
- ***Please also cite***: K. Zhao, S. Di, X. Liang, S. Li, D. Tao, J. Bessac, Z. Chen, and F. Cappello, "SDRBench: Scientific Data Reduction Benchmark for Lossy Compressors", International Workshop on Big Data Reduction (IWBDR2020), in conjunction with IEEE Bigdata20.
- ***Acknowledge***: the source of the dataset you used, the DOE NNSA ECP project, and the ECP CODAR project.
- ***Check***: the condition of publications (some dataset sources request prior check)
- ***Contact***: the compressor authors to get the correct compressor configuration according to each dataset and each comparison metrics.
- ***Dimension***: the order of the dimensions shown in the 'Format' column of the table is in row-major order (aka. C order), which is consistent with well-known I/O libraries such as HDF5. For example, for the CESM-ATM dataset (1800 x 3600), 1800 is higher dimension (changing slower) and 3600 is lower dimension (changing faster). For most compressors (such as SZ, ZFP and FPZIP), the dimensions should be given in the reverse order (such as -2 3600 1800) for their executables. If you are not sure about the order of dimension, one simple method is trying different dimension orders and selecting the results with highest compression ratios.

Data sets:

Name	Type	Format	Size (data)	Command Examples	Link
CESM-ATM *Source:* *Mark Taylor (SNL)*	Climate simulation	Dataset1 : 79 fields: 2D, 1800 x 3600 ; Dataset2 : 1 field : 3D, 26x1800x3600. Both are single precision, binary | 8 bit Entropy | 32 bit Entropy avg | 6.5 | 19.8 min | 1.5 | 2.3 max | 7.8 | 22.6	Dataset1 (raw): 1.47GB Dataset1 (cleared): 1.47GB Dataset2: 17GB (cleared data zeroed all background data)	**SZ(Compress)**: sz -z -f -i CLDHGH_1_1800_3600.f32 -M REL -R 1E-2 -2 3600 1800 **SZ(Decompress)**: sz -x -f -i CLDHGH_1_1800_3600.f32 -2 3600 1800 -s CLDHGH_1_1800_3600.f32.sz -a **ZFP**: zfp -f -i CLDHGH_1_1800_3600.f32 -z CLDHGH_1_1800_3600.f32.zfp -o CLDHGH_1_1800_3600.f32.zfp.out -2 3600 1800 -a 1E-2 -s **LibPressio:** pressio -b compressor=$COMP -i CLDHGH_1_1800_3600.f32 -d 3600 -d 1800 -t float -o rel=1e-2 -m time -m size -M all where $COMP can be sz, zfp, sz3, mgard, etc… **Z-checker-installer**: ./runZCCase.sh -f REL CESM-ATM raw-data-dir f32 3600 1800	Dataset1 (raw) Dataset1 (cleared) Dataset2 (raw) Metadata Dataset1's property Dataset2's property
EXAALT *Source:* *EXAALT team* *This dataset has been approved for unlimited release by Los Alamos National Laboratory and has been assigned LA-UR-18-25670.*	Molecular dynamics simulation	6 fields: x,y,z,vx,vy,vz, Each field stored separately, Single precision, Binary, Little-endian Data1 | 8 bit Entropy | 32 bit Entropy avg | 7.3 | 19.9 min | 7 | 18.4 max | 7.4 | 20.4 Data2 | 8 bit Entropy | 32 bit Entropy avg | 6.7 | 15.4 min | 6 | 9.40 max | 7.1 | 20.1 Data3 | 8 bit Entropy | 32 bit Entropy avg | 6.9 | 16.3 min | 6.8 | 15.6 max | 7.2 | 17.6	Dataset1: 60 MB Dataset2: 973 MB Dataset3: 2.4 GB	**SZ(Compress)**: sz -z -f -i xx.f32 -M REL -R 1E-2 -1 2869440 **SZ(Decompress)**: sz -x -f -i xx.f32 -1 2869440 -s xx.f32.sz -a **ZFP**: zfp -f -i xx.f32 -z xx.f32.zfp -o xx.f32.zfp.out -1 2869440 -a 1E-2 -s **LibPressio:** pressio -b compressor=$COMP -i xx.f32 -d 2869440 -t float -o rel=1e-2 -m time -m size -M all where $COMP can be sz, zfp, sz3, mgard, etc… **Z-checker-installer**: ./runZCCase.sh -f REL EXAALT raw-data-dir f32 2869440	Dataset1 Metadata1 Property1 Dataset2 Metadata2 Property2 Dataset 3 Metadata3 Property3
Hurricane ISABEL *Source:* *http://vis.computer.org/vis2004contest/data.html*	Weather simulation	13 fields: 3D, 100x500x500, single-precision, binary (cleared dataset by replacing background by 0) | 8 bit Entropy | 32 bit Entropy avg | 5.9 | 18.9 min | 0.7 | 1.3 max | 7.5 | 24.2	1.25GB	**SZ(Compress)**: sz -z -f -i Pf48.bin.f32 -M REL -R 1E-2 -3 500 500 100 **SZ(Decompress)**: sz -x -f -i Pf48.bin.f32 -3 500 500 100 -s Pf48.bin.f32.sz -a **ZFP**: zfp -f -i Pf48.bin.f32 -3 500 500 100 -z Pf48.bin.f32.zfp -o Pf48.bin.f32.zfp.out -a 1E-2 -s **LibPressio:** pressio -b compressor=$COMP -i Pf48.bin.f32 -d 500 -d 500 -d 100 -t float -o rel=1e-2 -m time -m size -M all where $COMP can be sz, zfp, sz3, mgard, etc…	Dataset Metadata Property



https://github.com/CODARcode/Z-checker


Z-checker-installer
Z-checker
Data Visualization Engine
Z-server
Output Engine
Analysis Kernel
Data Property Analyzer
Compression Checker
Input Engine
3rd-party Connecter
Data Source (stream, file, etc.) with formats (HDF5, NetCDF, ADIOS, binary)
3rd-party Library (R, FTTW)
Configuration Parser
Libpressio
Lossy Compressors (SZ, ZFP, FPZIP, MGARD, Jpeg, etc.)

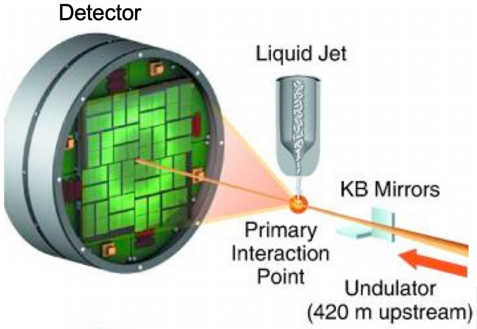
Argonne
NATIONAL LABORATORY

Putting all Together example: LCLS/Crystallography

Example of Success Story: Crystallography

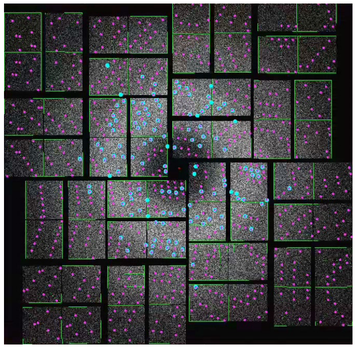
With Chuck Yoon: Stanford

1: X-ray Beam



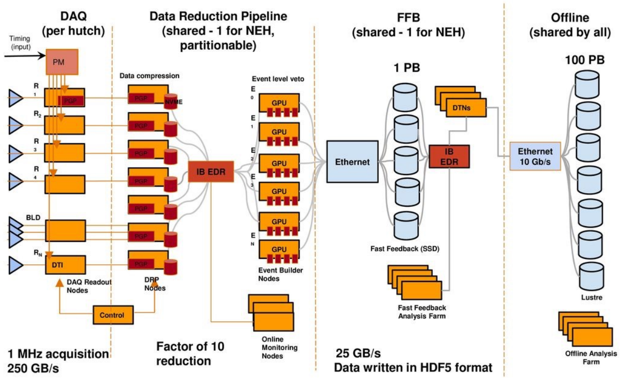
Diffraction before destruction
Number of pulses/sec: 120
Millions of diffraction patterns from crystals

2: Diffraction



3: Reduction

LCLS-II Data System



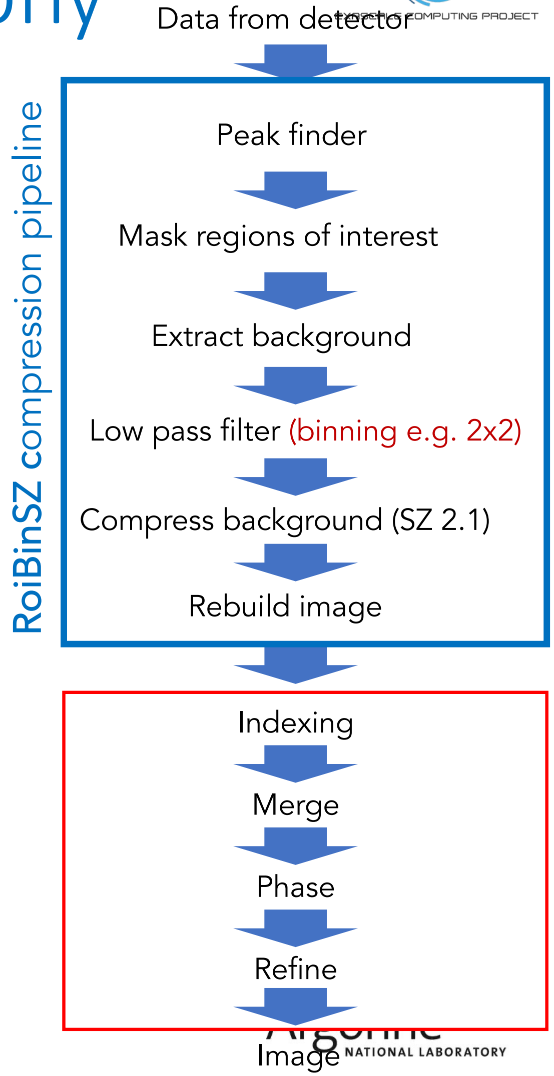
Context: LCLS II. Goal: Definition of reduction method

Detector produces:

- 2D images @ 250GB/s
- 4M pixel/event unsigned integers, in binary XTC2 format

Compression objectives: **CR of 10 or more** with error bound @ 500 MB/s/core

→ **RoiBinSZ** algorithm (regions of interest extraction + background binning + SZ background compression)



RoiBinSZ compression pipeline

Data from detector

Peak finder

Mask regions of interest

Extract background

Low pass filter (binning e.g. 2x2)

Compress background (SZ 2.1)

Rebuild image

Indexing

Merge

Phase

Refine

Image

First Level of Analysis Distortion: Indexing

Roibin SZ on Se-SAD SFX Dataset (Selenium)

selenobiotinyl-streptavidin on a cspad detector

Chuck Yoon: Stanford

	Original	Riobin SZ
Total compression ratio	1	**70.65**
Number of hits	744,150	744,150
Number indexed	255,065	255,918
Rsplit ↓	7.58%	**7.08%**
CC1/2 ↑	0.997	**0.997**
CCano ↑	0.087	**0.104**
Rwork ↓	0.206	**0.199**
Rfree ↓	0.231	**0.223**
Map-model CC ↑	**0.81**	0.8

↑: higher the better ↓: lower the better

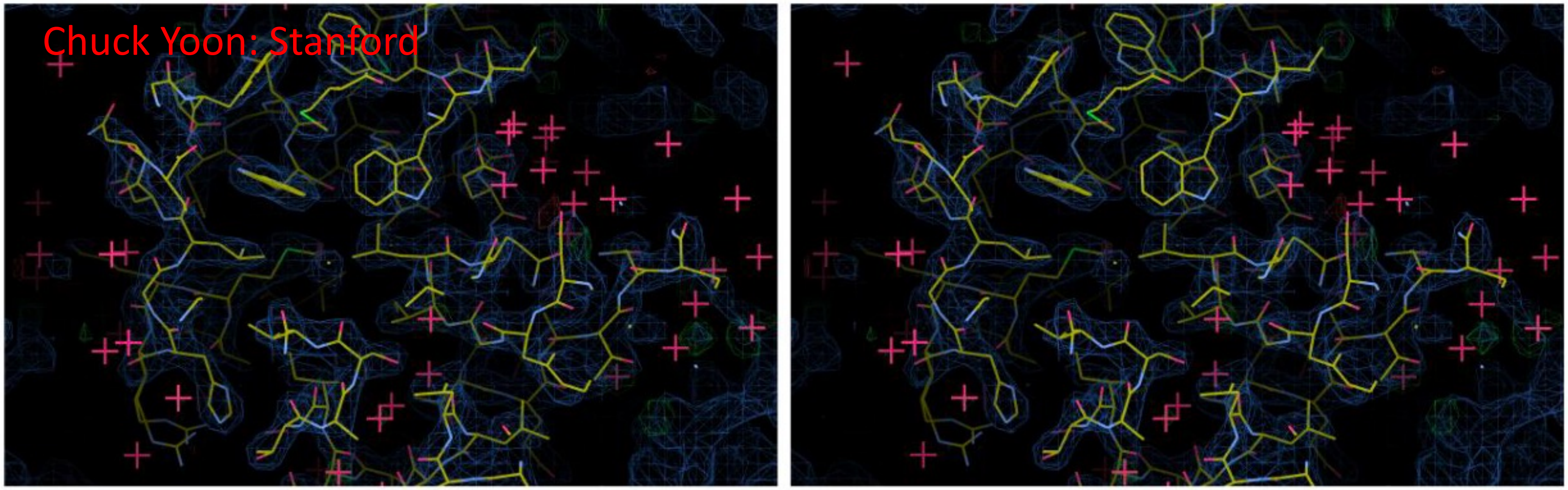
- Number of hits: An image with at least 15 peaks is considered a hit
- Number indexed: Number of crystals extracted from hits
- Rsplit: measure precision of averaged intensities/amplitudes
- CCano: The correlation coefficient of the Bijvoet differences of acentric reflections
- CC1/2: Pearson correlation coefficient.
- Rwork: measure of the agreement between the crystallographic model and the experimental X-ray diffraction data
- Rfree: Rwork computed on a small, random sample of data
- Map-model CC: cross-correlation between electron density map and model.

Final Level of Analysis Distortion: Protein Reconstruction

Lysozyme on a jungfrau4m detector

Reconstruction of Electron Densities Lysozyme

Very important role in our immune system: breaks up (digests) components of the cell walls of bacteria.



Chuck Yoon: Stanford

(a) original

(b) roibin-sz

The data on the right is 196x smaller (or 631× if also using Non-Hit Rejection)

What’s Next

- Compressing Memory/ Communications
  - Panda’s paper at IPDPS21: MVAPICH+ZFP. Collaboration with UTK on SZ for “mix precision” computation. Difficult problem: how for formulate the impact of lossy compression error on execution results?

- Feature Preservation
  - Preservation of derivatives, Structures (Blobs, Halos, Critical points, etc.), Reduction of artifacts, coupling with feature detections. Difficult problem: how to formulate/design error control from quality requirement on features

- Automation of Compression Pipeline Construction
  - Many possibilities of compression stages association (pre-processing, decorrelation, quantization, encoding). Difficult problem: How to automatically identify the best pipeline responding to user defined constraints in compression ratio, accuracy, speed

- **Compressibility Bounds**

Lossy Compressibility Bounds

Estimate “absolute” compression bounds →
compression ratio that no compressor can exceed given a user defined quality constraints: e.g. local max absolute/relative error

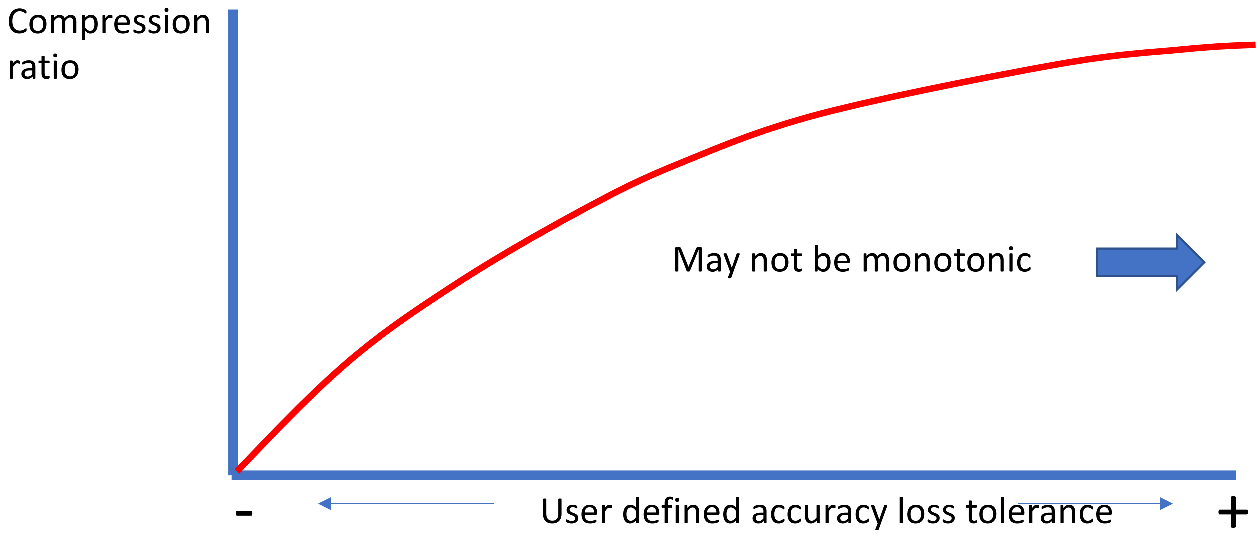
Why does this matter:

- We cannot know if current compressors are really good or not

Important: Scientific datasets in general can be considered as:
Non-Gaussian (distribution), with Memory (correlations), non-Stationary, non-Ergodic (do not visit all possible state) random processes

Objective → Roofline of lossy compressibility

- Lossy compressibility depends on user defined accuracy loss tolerance
- We want a compressibility estimation that capture correlations

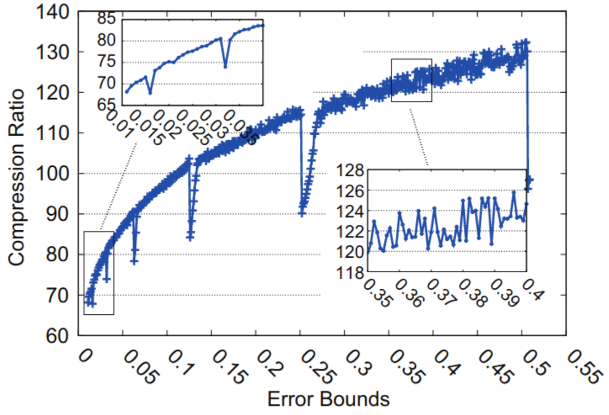


Compression ratio

May not be monotonic

- User defined accuracy loss tolerance +

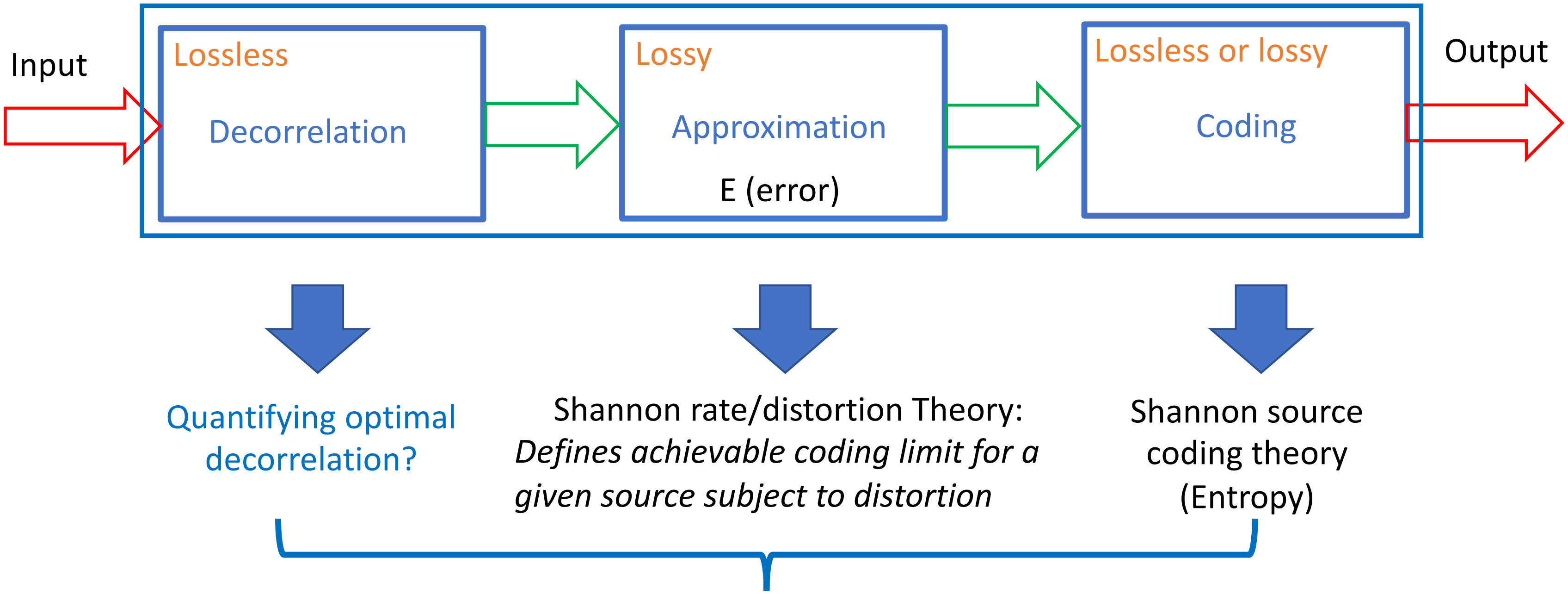
SZ actual roof line
Hurricane/QCLOUDf



- Should be specific to each dataset and user defined accuracy loss tolerance criteria
- Should be independent of existing compressors

Lossy Compression Pipeline

Typical design of a lossy compressor for scientific data



Quantifying optimal decorrelation?

Shannon rate/distortion Theory: *Defines achievable coding limit for a given source subject to distortion*

Shannon source coding theory (Entropy)

Formulation of Compressibility Bound

Bound on Lossy Compressibility (Absolute: even if not practical)?

Many researches/results in information theory extending the Shannon rate/distortion theory.

→ But assumptions do not correspond to Scientific data compression: e.g. Gaussian source with memory or Non-gaussian memoryless, non-stationary Gaussian, etc.

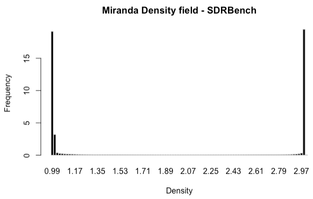
Let's look specifically at the decorrelation stage:

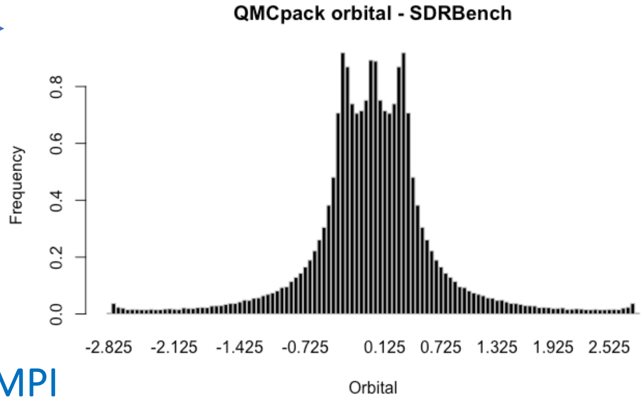
Both produce optimal decorrelation WRT coding gain, in Gaussian case. However, assumptions and data overhead limit their applicability.

KLT (Karhunen–Loève transform) is know to be optimal WRT for decorrelation efficiency (given a bit budget in the compressed format, KTL minimizes the distortion)

But KLT is not optimal (transform) for scientific datasets

- → Because the data distribution is not Gaussian
- → Only for transform based compressors
- → Data needs to be homoscedastic (same variance everywhere on the field)
- → Intrinsically needs a notion of blocks to compute the covariance





SVD (and HoSVD: Tucker decomposition): TTRESH and Tucker-MPI

- → Not recommended for Images (DCT performs better), but works well for Vis. ≥3D
- → There is not yet proper consensus on a mathematical framework to select the core tensor elements for optimal truncation
- → Needs to identify an elimination strategy on coefficient to coefficient basis (truncation strategy)

Lossy Compressibility Bounds

A less difficult problem:

Estimate compression/quality for several compressors. →
Compression bound WRT existing compressors

Why does this matter:

- **Enable fast, automated configuration of a single compressor** to have max quality that will fit in available storage [7]. Cosmology simulations [8], climate simulations [9] and X-ray crystallography [10]
- **Enable quickly choosing between several compressors** with highest CR at runtime in order to minimize data size.
- **Accurately pre-allocate memory** when using compression to expand the amount of on-node logical memory to run applications that utilize in-line compression [11] and quantum chemistry simulations [12].
- **Accurately foreseeing the data transfer time** of I/O or on networks across different devices or sites when compression is used to optimize resource utilization across network links when streaming data [13].

Bound on Lossy Compressibility (compressor)?

Existing prediction models use knowledge of compressor designs (white box).
→ Not accurate enough (prediction err can be >100%)

- **Formulation of a generic (compressor free) statistical prediction model.**
- **Use training from observed compression ratios (black box) for its specialization to a compressor.**

2 steps:
Step 1: Use Statistical predictors on dataset:
a) SVD to exploit spatial correlation, b) standard deviation to account for variability, c) quantized entropy to represent lossyness and codding
Step 2: Train compression models from regressions (linear and spline-based) to fit predictors to actual observations

Can be applied and specialized to any lossy compressor before compression

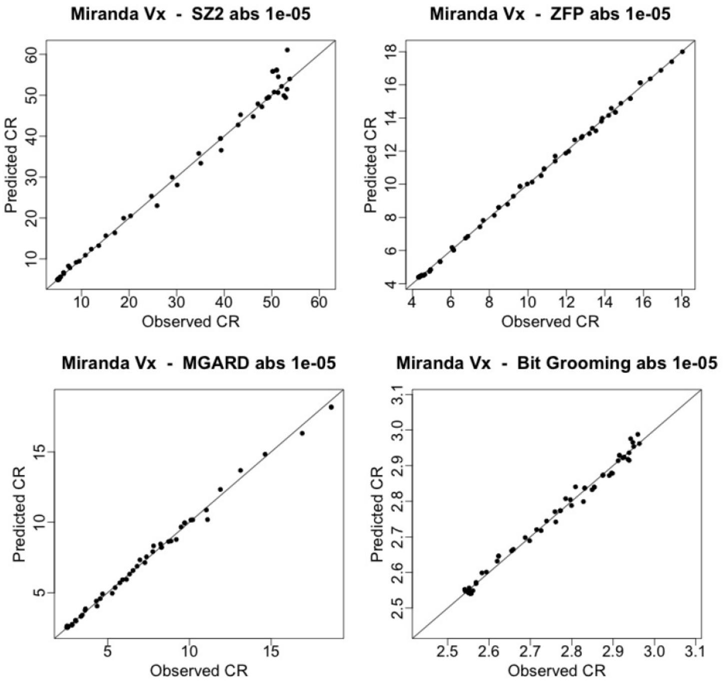
Linear model:

$$\begin{aligned}\overset{*}{\log}(\mathrm{CR}) &= a + b \times \log(\text{q-ent}) + c \times \log\left(\frac{\text{SVD-trunc}}{\sigma}\right)\\ &\quad + d \times \log(\text{q-ent}) \times \log\left(\frac{\text{SVD-trunc}}{\sigma}\right) + \epsilon, \qquad (1)\end{aligned}$$

where ϵ is a Gaussian random variable with mean 0 and standard deviation σ_{eps}. Coefficients a, b, c, d and σ_{eps} are estimated by least-square estimation with the R-function lm.

Tested on 4 different Compressors:
(out-of-sample prediction)

-SZ (prediction)
-ZFP (Transform)
-MGARD (Multi-grid)
-Bit Grooming (truncation and bit operations)



*Uses log because values may appear from dividing by the standard deviation

Conclusion

Lossy Compression for scientific data:

- A very active research topic
- Many teams working on the topic
- Excellent progress in the past 5-6 years
- Significant interest/adoption by apps
- Still many interesting open questions

Thanks

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

RESEARCH SPONSORED BY

The Exascale Computing Project

A Collaborative effort of the U.S. Department of Energy Office of Science And the National Nuclear Security Administration

(17-SC-20-SC)

U.S. DEPARTMENT OF ENERGY | Office of Science

NNSA National Nuclear Security Administration