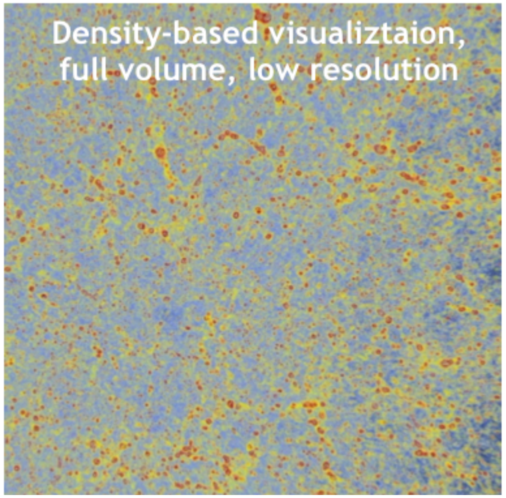
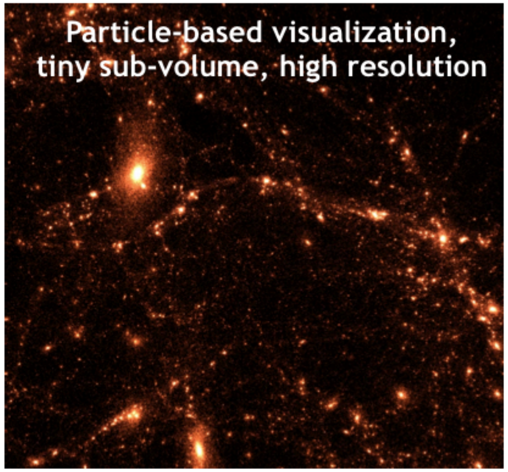
SZ-1.4: Significantly Improving Lossy Compression for Scientific Data Sets Based on Multidimensional Prediction and Error-Controlled Quantization

Dingwen Tao (University of California, Riverside)
Sheng Di (Argonne National Laboratory)
Zizhong Chen (University of California, Riverside)
Franck Cappello (Argonne National Laboratory & UIUC)

Introduction (1)

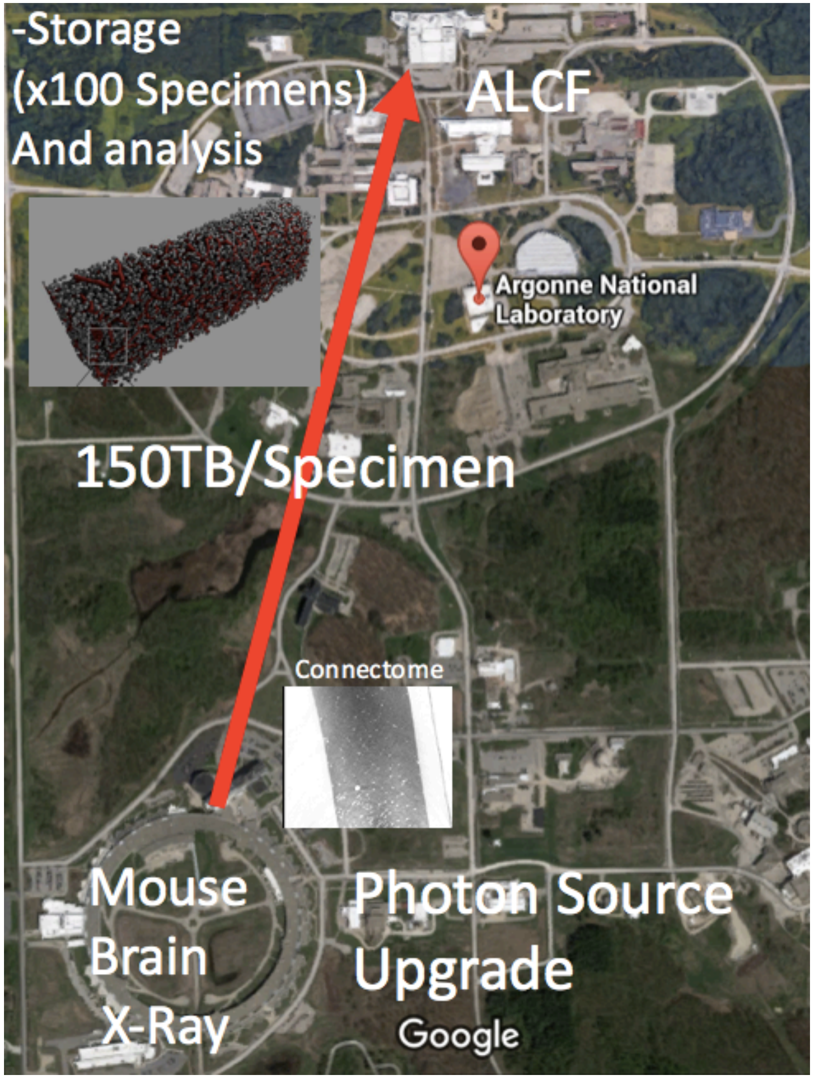
- Extremely large amount of data are produced by scientific simulations and instruments
  - HACC (Cosmology Simulation)
    - 20 PB data: a single 1-trillion-particle simulation
    - Mira at ANL: 26 PB file system storage
    - 20 PB / 26 PB ~ 80%
  - CESM/CMIP5 (Climate Simulation)
    - 2.5 PB raw data produced
    - 170 TB post-processed data





Two partial visualizations of HACC simulation data: coarse grain on full volume or full resolution on small sub-volumes

Introduction (2)



- APS-U: next-generation APS (Advanced Photon Source) project at ANL
  - 15 PB data for storage
  - 35 TB post-processed floating-point data
  - 100 GB/s bandwidth between APS and Mira
  - 15 PB / 100 GB/s ~ 10^5 seconds (42 hours)
- **Data compression provides a promising way to relieve I/O and storage pressure!!**

Motivation – Limitations of Existing Lossless Compressors

- Existing lossless compressors work **not efficiently** on large-scale scientific data (compression ratio up to 2)

Table 1: Compression ratios for lossless compressors on large-scale simulations

	bzip2	dfcm	fsd	gzip	lzpx	p7zip	rar	zzip
aztec	1.15	1.69	1.42	1.22	1.15	1.39	1.26	1.15
bt	1.10	1.36	1.02	1.13	1.10	1.32	1.15	1.10
eulag	1.04	1.23	1.06	1.06	1.05	1.15	1.07	1.09
lu	1.02	1.23	0.99	1.05	1.03	1.22	1.07	1.03
sp	1.08	1.25	0.95	1.11	1.07	1.31	1.14	1.07
sppm	6.78	4.16	2.14	6.31	7.94	8.31	7.68	---
sweep3d	1.06	1.49	1.20	1.09	1.19	1.26	1.30	1.35
geo_mean	1.40	1.60	1.20	1.42	1.46	1.66	1.52	1.13

Compression ratio = Original data size / Compressed data size

Outline

- Introduction
  - Large amount of scientific data
  - Limitations of lossless compression
- Existing lossy compressors and limitations

Outline

- Introduction
  - Large amount of scientific data
  - Limitations of lossless compression
- Existing lossy compressors and limitations
- Our Designs
  - Multidimensional / Multilayer Prediction Model
  - Adaptive Error-Controlled Quantization

Outline

- Introduction
  - Large amount of scientific data
  - Limitations of lossless compression
- Existing lossy compressors and limitations
- Our Designs
  - Multidimensional / Multilayer Prediction Model
  - Adaptive Error-Controlled Quantization
- Metrics and Measurements

Outline

- Introduction
  - Large amount of scientific data
  - Limitations of lossless compression
- Existing lossy compressors and limitations
- Our Designs
  - Multidimensional / Multilayer Prediction Model
  - Adaptive Error-Controlled Quantization
- Metrics and Measurements
- Empirical Evaluation
  - Compression performance & Parallel evaluation
- Conclusion

Outline

- Introduction
  - Large amount of scientific data
  - Limitations of lossless compression
- Existing lossy compressors and limitations
- Our Designs
  - Multidimensional / Multilayer Prediction Model
  - Adaptive Error-Controlled Quantization
- Metrics and Measurements
- Empirical Evaluation
  - Compression performance & Parallel evaluation
- Conclusion

Existing Lossy Compressors

- Existing state-of-the-art lossy compressors
  - ***ISABELA*** (NCSU)
    - Sorting preconditioner
    - B-Spline interpolation
  - ***ZFP*** (LLNL)
    - Customized orthogonal block transform
    - Exponent alignment
    - Block-wise bit-stream truncation
  - ***SZ-1.1*** (ANL)
    - Linear and quadratic 1D curve fitting for prediction
    - Binary representation analysis for unpredictable data
  - Others: non-competitive (as shown in SZ-1.1 paper – IPDPS'16)

Limitations of Existing Lossy Compressor

- ***ISABELA***
  - Sorting is very time-consuming
  - Storing initial index extremely limits compression ratio
- ***ZFP***
  - Over-preserves errors in decompressed data with respect to user-set error bound
  - Might not respect strictly error bounds in some extreme cases due to exponent alignment step (see details in the paper)
  - Not effective on low dimensional data sets (e.g., 1D and 2D)
- ***SZ-1.1***
  - Prediction: only adopts 1D prediction model, i.e., linear / quadratic curve fitting
  - Quantization: prediction-hitting rate drops quickly when data are not smooth or high-accuracy requirement

Outline

- Introduction
  - Large amount of scientific data
  - Limitations of lossless compression
- Existing lossy compressors and limitations
- Our Designs
  - Multidimensional / Multilayer Prediction Model
  - Adaptive Error-Controlled Quantization
- Metrics and Measurements
- Empirical Evaluation
  - Compression performance & Parallel evaluation
- Conclusion

SZ-1.4: Significantly Improving Error-bounded Lossy Compressor

- The whole compression procedure
  1. Point-wise **multidimensional / multilayer data prediction**
  2. **Error-bounded quantization** (linear-scaling quantization)
  3. Variable-length encoding (customized Huffman encoding)
  4. Unpredictable data compression (similar to SZ-1.1)
  5. Dictionary-based encoding (customized LZ77) (optional)

Algorithm 1 Proposed Lossy Compression Algorithm Using Multi-layer Prediction and AEQVE Model

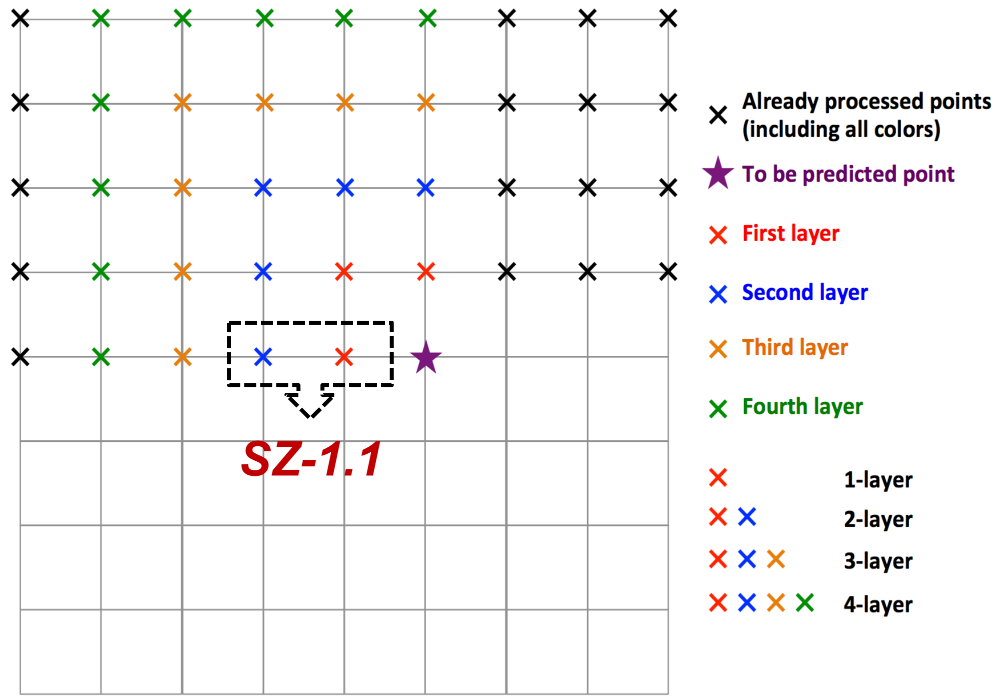
Input: d-dimensional array data set X of the size $n^{(1)} \times n^{(2)} \times \cdots \times n^{(d)}$, error bound mode (absolute and/or value-range-based relative error bound), user-set error bound eb_{abs}/eb_{rel}.

Output: Compressed byte data C_{quan}, D_{unpred}

Set $eb = eb_{abs}$, or $eb_{rel} \cdot R_X$, or $\min\{eb_{abs}, eb_{rel} \cdot R_X\}$
Set θ and m
Compute $(n+1)^d - 1$ coefficients of n-layer prediction **O(1)**
for $i^{(d)} = 1 \cdots n^{(d)}$ **do**
  for $i^{(d-1)} = 1 \cdots n^{(d-1)}$ **do**
    $\cdots$
    for $i^{(1)} = 1 \cdots n^{(1)}$ **do**
      /* Process data point $X_{i^{(1)} \dots i^{(d)}}$ */
      Compute predicted value V_{pred} using n-layer prediction model for $X_{i^{(1)} \dots i^{(d)}}$ **O(1)**
      Compute $Diff = V_{pred} - V_{i^{(1)} \dots i^{(d)}}$, where $V_{i^{(1)} \dots i^{(d)}}$ is the original value of $X_{i^{(1)} \dots i^{(d)}}$ **O(1)**
      Encode $X_{i^{(1)} \dots i^{(d)}}$ using 2^m quantization codes and store the quantization code to C_{quan} **O(1)**
      if $(|Diff|/eb > 2^m - 1)$ **then**
        /* Process unpredictable data */
        Compress $V_{i^{(1)} \dots i^{(d)}}$ using binary-representation analysis and store to D_{unpred} **O(1)**
      end if
      Compute and record decompressed value of $X_{i^{(1)} \dots i^{(d)}}$ **O(1)**
    end for
    $\cdots$
  end for
end for
Compress C_{quan} using variable-length encoding **O(N)**
Count number of predictable data points and compute prediction hitting rate R_{PH} **O(N)**
if $(R_{PH} < \theta)$ **then**
  Give the user a suggestion: increasing # of quantization intervals
end if

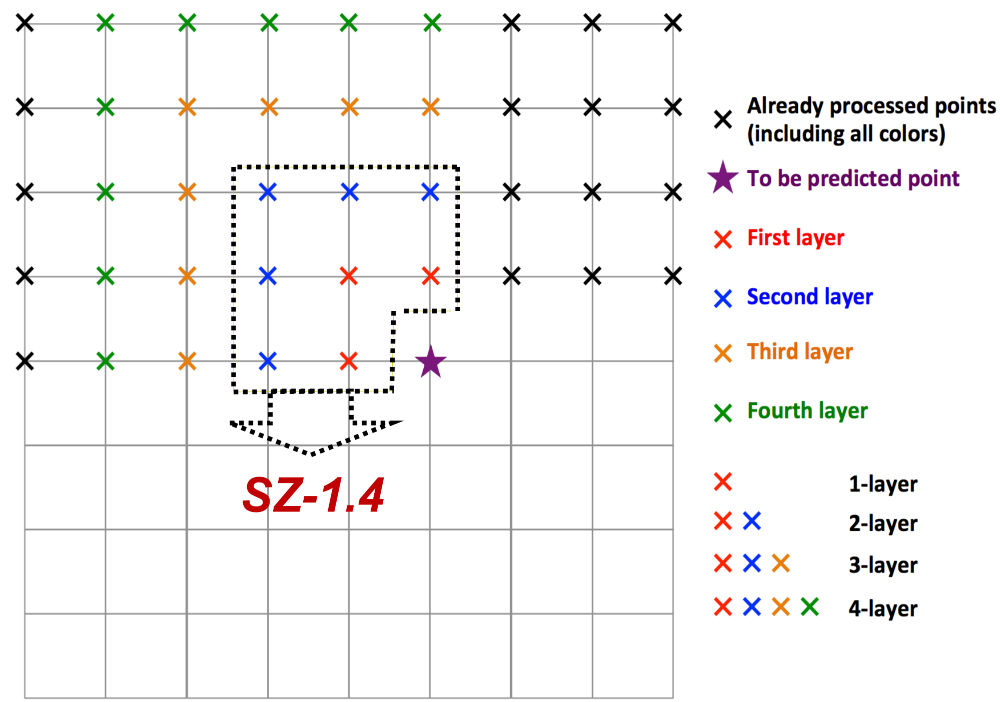
Our Designs – Multidimensional / Multilayer Prediction Model (1)

- Use 2D data set as an example
- Suppose **purple** star is data point to be predicted
- ***SZ-1.1***'s prediction model
  - Only use 1D information in prediction



Our Designs – Multidimensional / Multilayer Prediction Model (1)

- Use 2D data set as an example
- Suppose purple star is data point to be predicted
- ***SZ-1.1***'s prediction model
  - Only use 1D information in prediction
- ***SZ-1.4***'s prediction model
  - Multidimensional prediction – use adjacent data points along *multiple directions*
  - Multilayer prediction – use adjacent data points in *multiple layers* (e.g., 2-layer includes red + blue points)

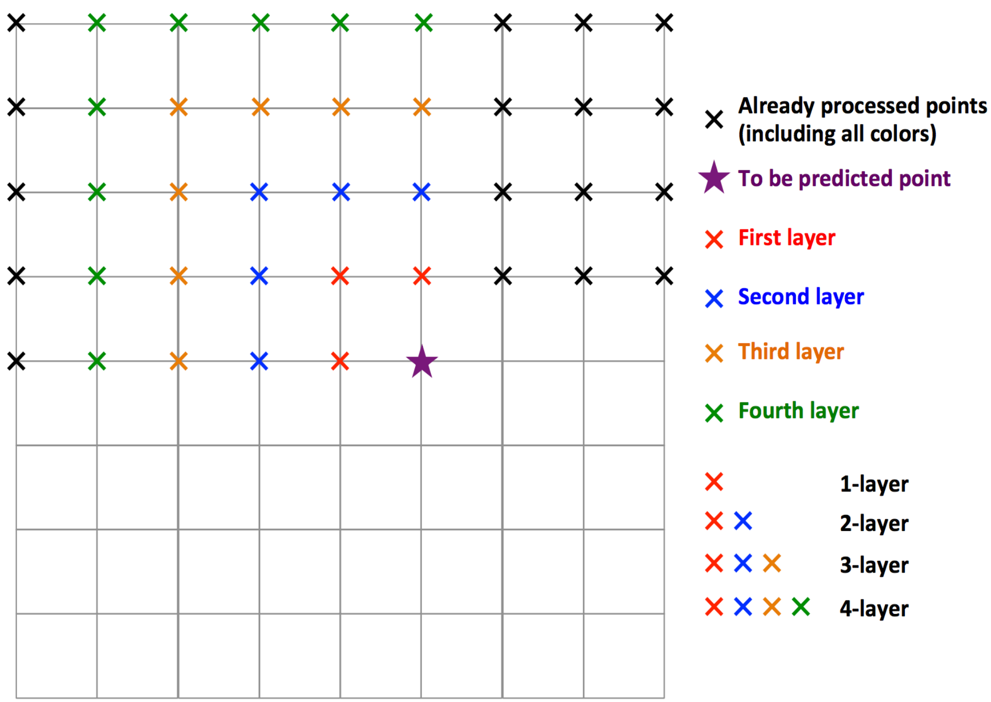


Our Designs – Multidimensional / Multilayer Prediction Model (2)

- Target: use *n*-layer prediction
- Point to be predicted: **(i_0, j_0)**
- Construct a fitting surface *f(x, y)* based (i_0, j_0) 's adjacent points

$$f(x,y) = \sum_{0 \le i+j \le 2n-1}^{i,j \ge 0} a_{i,j} x^i y^j$$

  - n(2n+1) unknown coefficients
- Straightforward idea to get *f(x, y)*
  - Choose n(2n+1) data points
  - Assume fitting surface go through all n(2n+1) points
  - Solve unknown coefficients
- Problem: not any n(2n+1) points can be on *f(x, y)* at the same time



Our Designs – Multidimensional / Multilayer Prediction Model (3)

- **Theorem**
  - The n(2n+1) points – $\boxed{\{(k_1,k_2) \mid 0 \le k_1+k_2 \le 2n-1,\ k_1,\ k_2 \ge 0\}}$ – can be used for solving the n(2n+1) unknown coefficients in *f(x, y)*
  - Fitting surface's value on point (i_0, j_0), $f(i_0, j_0)$, can be expressed explicitly by the n(2n+1) points' values

$$f(i_0, j_0) = \sum_{0 \le k_1, k_2 \le n}^{(k_1,k_2) \ne (0,0)} (-1)^{k_1+k_2+1} \binom{n}{k_1} \binom{n}{k_2} V(i_0 - k_1, j_0 - k_2) \tag{10}$$

- $f(i_0, j_0)$ serves as the prediction value for point (i_0, j_0) , i.e., Equation (10)
  - Note V(i, j) is the **decompressed value** of point (i, j)
- Our model can utilize different number of layers (i.e., *n*) in prediction – **multidimensional / multilayer prediction model**
- Default setting in SZ-1.4
  - Using 1-layer prediction (n = 1)
  - $f(i_0, j_0) = V(i_0, j_0-1) + V(i_0-1, j_0) – V(i_0-1, j_0-1)$

Our Designs – Multidimensional / Multilayer Prediction Model (3)

- **Theorem**
  - The n(2n+1) points – $\{(k_1,k_2) \mid 0 \le k_1+k_2 \le 2n-1,\ k_1, k_2 \ge 0\}$ – can be used for solving the n(2n+1) unknown coefficients in *f(x, y)*
  - Fitting surface's value on point (i_0, j_0), $f(i_0, j_0)$, can be expressed explicitly by the n(2n+1) points' values

$$\boxed{f(i_0, j_0) = \sum_{0 \le k_1, k_2 \le n}^{(k_1,k_2) \ne (0,0)} (-1)^{k_1+k_2+1} \binom{n}{k_1} \binom{n}{k_2} V(i_0 - k_1, j_0 - k_2)} \tag{10}$$

- $f(i_0, j_0)$ serves as the prediction value for point (i_0, j_0) , i.e., Equation (10)
  - Note V(i, j) is the **decompressed value** of point (i, j)
- Our model can utilize different number of layers (i.e., *n*) in prediction – **multidimensional / multilayer prediction model**
- Default setting in SZ-1.4
  - Using 1-layer prediction (n = 1)
  - $f(i_0, j_0) = V(i_0, j_0-1) + V(i_0-1, j_0) - V(i_0-1, j_0-1)$

Our Designs – Multidimensional / Multilayer Prediction Model (4)

- Prediction of each data point is same
- Coefficients are computed before whole compression
- Computation complexity of prediction is *O(1)* for each point
- Relation with Lorenzo predictor
  - Equivalent to Lorenzo predictor when using 1-layer prediction (n = 1)
  - Our model is the generic expression

Table I
FORMULAS OF 1, 2, 3, 4-LAYER PREDICTION FOR TWO-DIMENSIONAL DATA SETS

	Prediction Formula
1-Layer	$f(i_0,j_0) = V(i_0,j_0-1) + V(i_0-1,j_0) - V(i_0-1,j_0-1)$
2-Layer	$f(i_0,j_0) = 2V(i_0-1,j_0) + 2V(i_0,j_0-1)$ $-4V(i_0-1,j_0-1) - V(i_0-2,j_0) - V(i_0,j_0-2)$ $+2V(i_0-2,j_0-1) + 2V(i_0-1,j_0-2) - V(i_0-2,j_0-2)$
3-Layer	$f(i_0,j_0) = 3V(i_0-1,j_0) + 3V(i_0,j_0-1)$ $-9V(i_0-1,j_0-1) - 3V(i_0-2,j_0) - 3V(i_0,j_0-2)$ $+9V(i_0-2,j_0-1) + 9V(i_0-1,j_0-2) - 9V(i_0-2,j_0-2)$ $+V(i_0-3,j_0) + V(i_0,j_0-3)$ $-3V(i_0-3,j_0-1) - 3V(i_0-1,j_0-3)$ $+3V(i_0-3,j_0-2) + 3V(i_0-2,j_0-3) - V(i_0-3,j_0-3)$
4-Layer	$f(i_0,j_0) = 4V(i_0-1,j_0) + 4V(i_0,j_0-1)$ $-16V(i_0-1,j_0-1) - 6V(i_0-2,j_0) - 6V(i_0,j_0-2)$ $+24V(i_0-2,j_0-1) + 24V(i_0-1,j_0-2)$ $-36V(i_0-2,j_0-2) + 4V(i_0-3,j_0) + 4V(i_0,j_0-3)$ $-16V(i_0-3,j_0-1) - 16V(i_0-1,j_0-3) + 24V(i_0-3,j_0-2)$ $+24V(i_0-2,j_0-3) - 16V(i_0-3,j_0-3)$ $-V(i_0-4,j_0) - V(i_0,j_0-4) + 4V(i_0-4,j_0-1)$ $+4V(i_0-1,j_0-4) - 6V(i_0-4,j_0-2) - 6V(i_0-2,j_0-4)$ $+4V(i_0-4,j_0-3) + 4V(i_0-3,j_0-4) - V(i_0-4,j_0-4)$

Outline

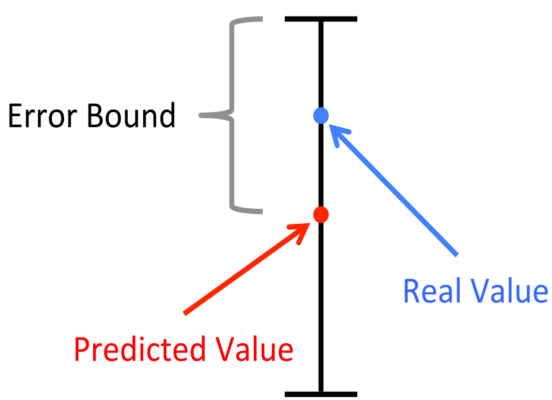
- Introduction
  - Large amount of scientific data
  - Limitations of lossless compression
- Existing lossy compressors and limitations
- Our Designs
  - Multidimensional / Multilayer Prediction Model
  - Adaptive Error-Controlled Quantization
- Metrics and Measurements
- Empirical Evaluation
  - Compression performance & Parallel evaluation
- Conclusion

Our Designs – Adaptive Error-Controlled Quantization (1)

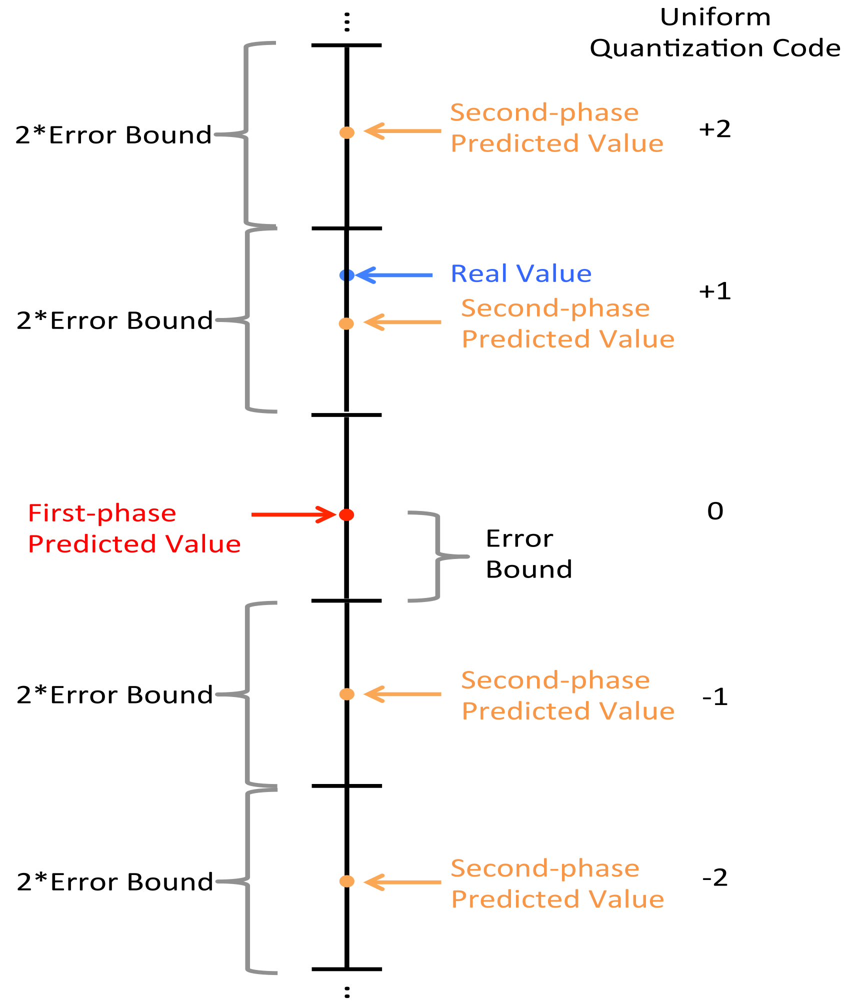
SZ-1.1 → SZ-1.4

(i) Expand quantization intervals from predicted value (made by previous prediction model) by ***linear scaling of the error bound***

(ii) Encode the real value using the quantization interval number (quantization code)



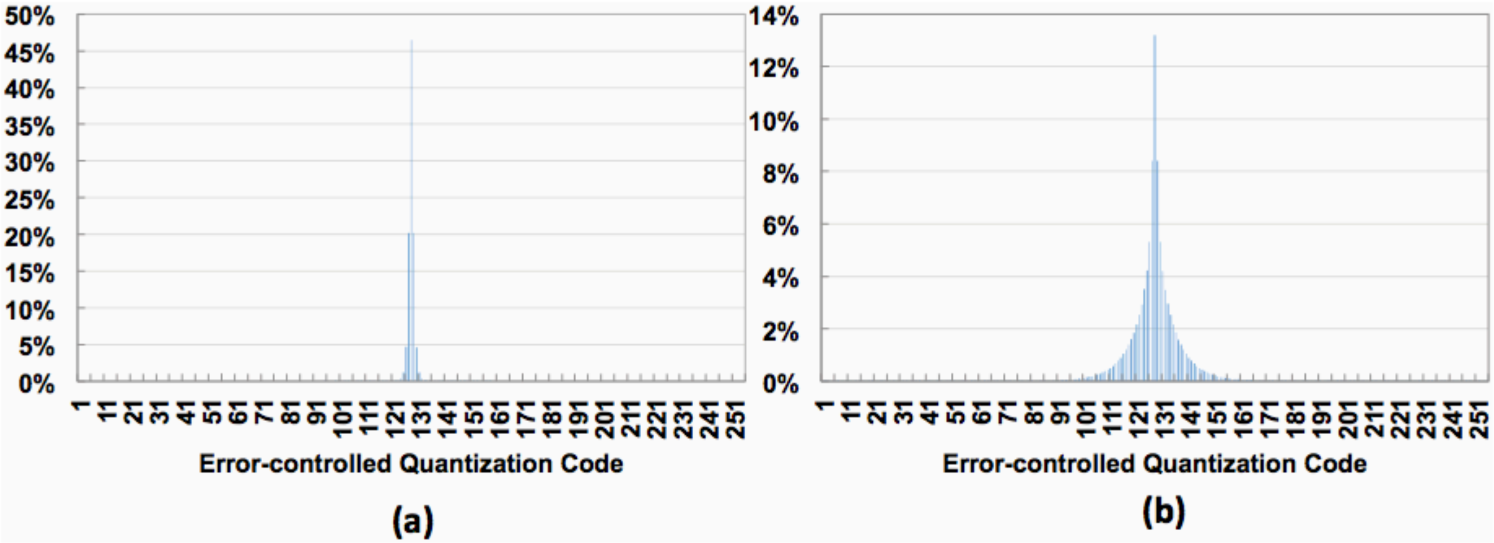
New design



Quantization with **one** interval in SZ-1.1

Quantization with multiple intervals (linear scaling) in SZ-1.4

Our Designs – Adaptive Error-Controlled Quantization (2)

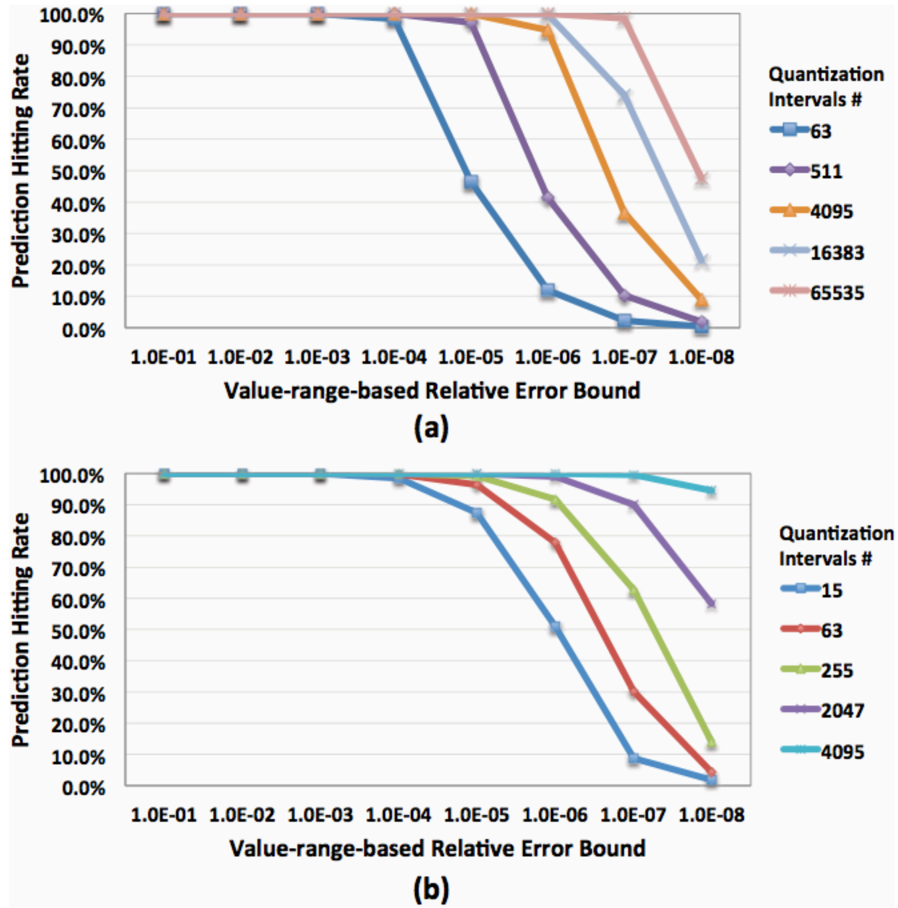


- Figure: distribution of quantization codes produced by error-controlled quantization encoder on climate simulation data (ATM) with two different error bounds and 255 quantization intervals (1 byte)
- Distribution: ***FAIRLY UNEVEN***
- We can further reduce the size of quantization codes by using ***variable-length encoding*** (e.g., **Huffman encoding**, arithmetic encoding)

Our Designs – Adaptive Error-Controlled Quantization (3)

- How many quantization intervals?
  - Excess: wasteful bits for quantization code
  - Insufficient: unable to cover irregular/spiky data
  - Unpredictable data: hard-to-compress, relatively larger than quantization code
- Adaptive # of quantization intervals to assure prediction-hitting rate > θ (θ is a threshold)

1. Sampling on initial data
2. Estimate quantization interval # for each sampling point
3. Count how many sampling points for fixed interval #
4. Sum numbers with increasing interval # until ratio of #covered_points / #total_points > θ
5. Take power of 2 for # of quantization intervals



(a)

(b)

Outline

- Introduction
  - Large amount of scientific data
  - Limitations of lossless compression
- Existing lossy compressors and limitations
- Our Designs
  - Multidimensional / Multilayer Prediction Model
  - Adaptive Error-Controlled Quantization
- Metrics and Measurements
- Empirical Evaluation
  - Compression performance & Parallel evaluation
- Conclusion

Measurements and Metrics (1)

- **Point-wise Compression Error**
  - Point-wise error e_i = | decompressed data – initial data | for data point i
  - User-set error bound *eb*
  - Error bounded: $e_i < eb$ for each point i
- **Compression ratio (CR)**
  - CR = Initial data size / compressed data size
- **Bit-rate (BR)**
  - Number of amortized bits per value
  - BR of initial floating-point data = 32 or 64
  - BR of compressed data = 32 (64) / CR
- **Compression / decompression speed**
  - B, MB, GB / Seconds

Measurements and Metrics (2)

- **Distortion**
  - Statistical error between initial and decompressed data
  - Commonly used metrics (based on L_2 norm)
    - Root mean squared error (RMSE)
    - Normalized root mean squared error (NRMSE)
    - Peak signal-to-noise ratio (PSNR)
  - $PSNR = -20 * \log_{10}(NRMSE)$
- **Rate-distortion**
  - For a fair comparison across fixed-rate (e.g., ZFP) and fixed-accuracy compressors (e.g., SZ-1.1/SZ-1.4)
  - Quality (distortion) per bit of compressed storage
  - e.g., PSNR / BR (dB/bit)
- **Autocorrelation of Compression Errors**

Outline

- Introduction
  - Large amount of scientific data
  - Limitations of lossless compression
- Existing lossy compressors and limitations
- Our Designs
  - Multidimensional / Multilayer Prediction Model
  - Adaptive Error-Controlled Quantization
- Metrics and Measurements
- Empirical Evaluation
  - Compression performance & Parallel evaluation
- Conclusion

Empirical Evaluation

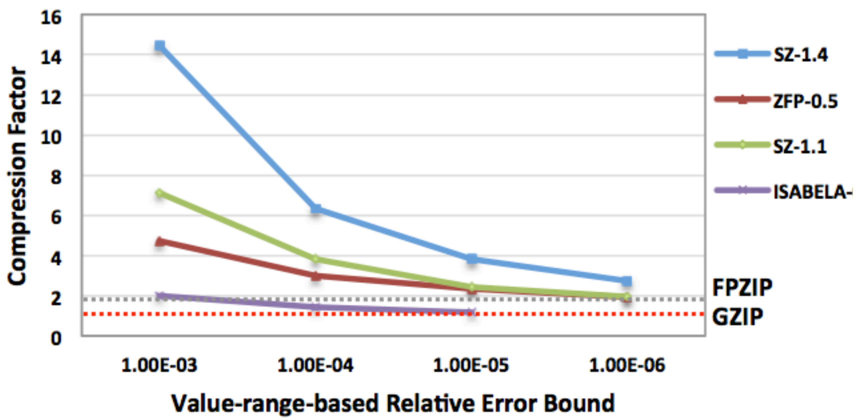
- Experimental platforms
  - Serial: iMac with 2.3 GHz Intel Core i7 + 32 GB DDR3 Memory
  - Parallel: Blues cluster at ANL – each node with 2 Intel Xeon E5-2670 processors + 64 GB DDR3 Memory
- Experimental data (single-floating point)
  - ATM: 2D data sets from climate/atmosphere simulations
  - APS: 2D data sets from X-ray scientific research
  - Hurricane: 3D data sets from hurricane Isabel simulation

Table III
DESCRIPTION OF DATA SETS USED IN EMPIRICAL PERFORMANCE EVALUATION

	Data Source	Dimension Size	Data Size	File Number
ATM	Climate simulation	1800×3600	2.6 TB	11400
APS	X-ray instrument	2560×2560	40 GB	1518
Hurricane	Hurricane simulation	$100 \times 500 \times 500$	1.2 GB	624

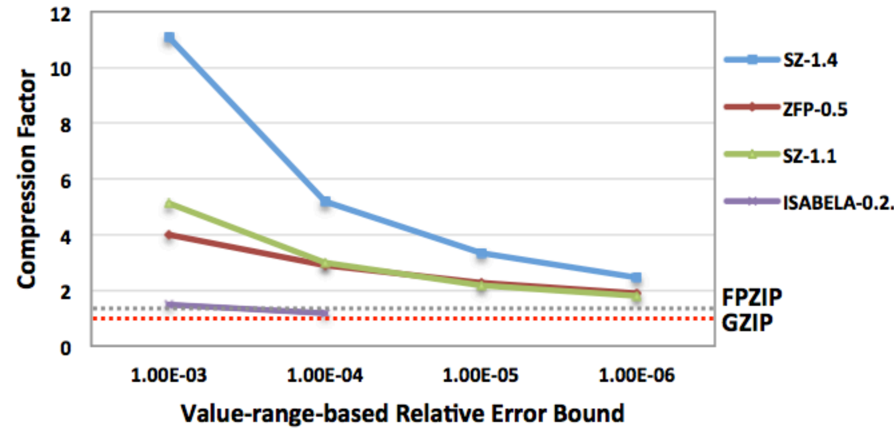
Empirical Evaluation – Compression Ratio (1)

ATM



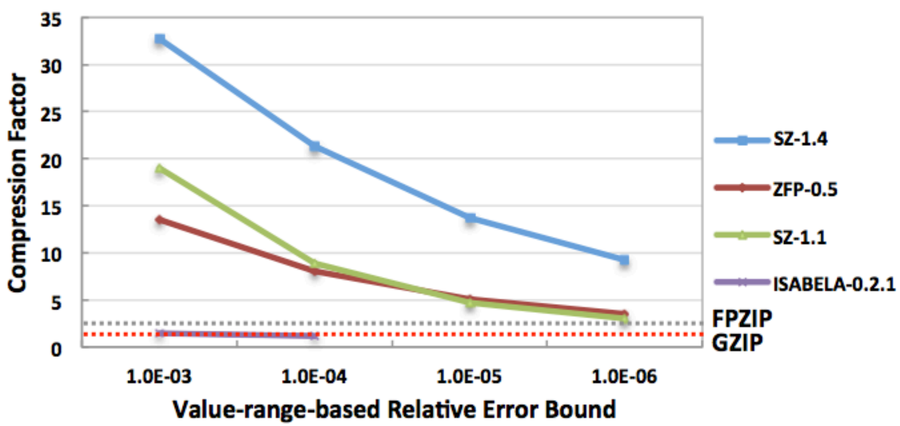
(a)

APS



(b)

Hurricane



(c)

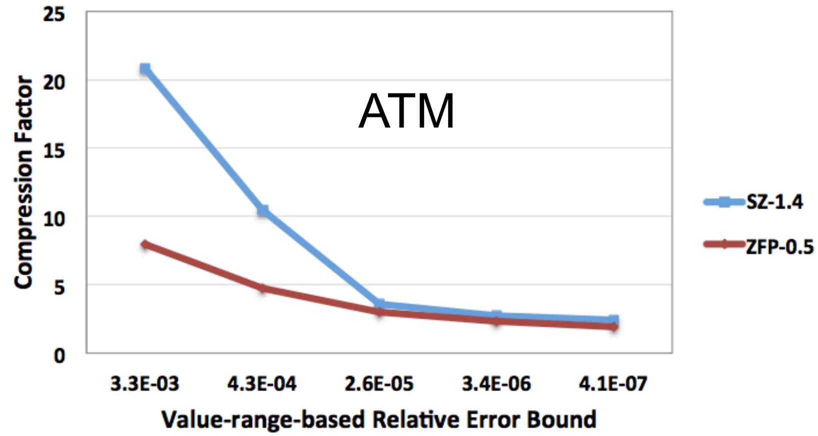
- Value-range-based (VRB) relative error bound = absolute error bound / data value range
- E.g., VRB relative error bound = 1E-4

1.9x ~ ZFP
2.2x ~ SZ-1.1

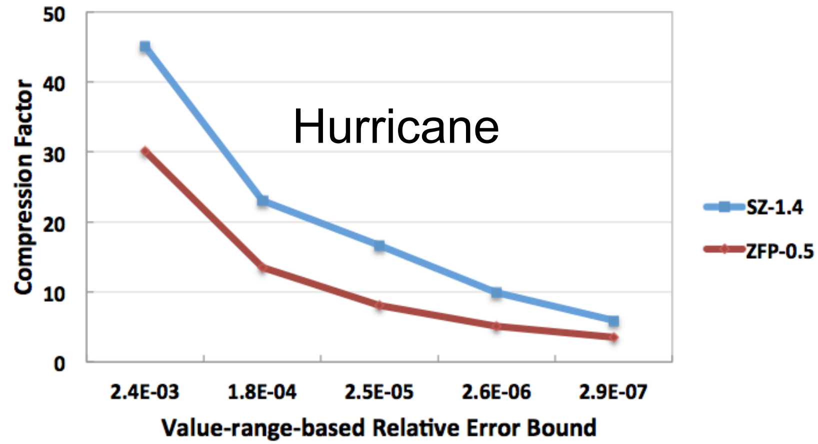
Empirical Evaluation – Compression Ratio (2)

Table V
MAXIMUM COMPRESSION ERRORS (NORMALIZED TO VALUE RANGE) USING SZ-1.4 AND ZFP WITH DIFFERENT USER-SET VALUE-RANGE-BASED ERROR BOUNDS

User-set eb_{rel}	**ATM**		**Hurricane**	
	SZ-1.4	*ZFP*	*SZ-1.4*	*ZFP*
10^{-2}	1.0×10^{-2}	3.3×10^{-3}	1.0×10^{-2}	2.4×10^{-3}
10^{-3}	1.0×10^{-3}	4.3×10^{-4}	1.0×10^{-3}	1.8×10^{-4}
10^{-4}	1.0×10^{-4}	2.6×10^{-5}	1.0×10^{-4}	2.5×10^{-5}
10^{-5}	1.0×10^{-5}	3.4×10^{-6}	1.0×10^{-5}	2.6×10^{-6}
10^{-6}	1.0×10^{-6}	4.1×10^{-7}	1.0×10^{-6}	2.9×10^{-7}



(a)



(b)

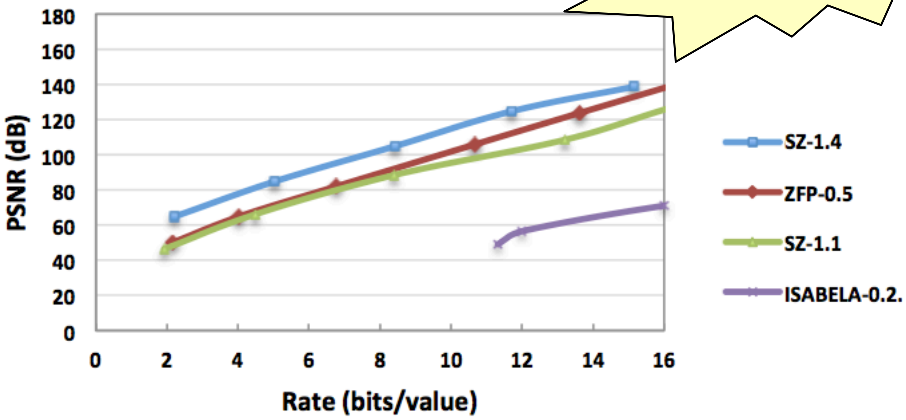
VRB relative *eb* around 1E-4

- **2.6x** of ZFP on ATM
- **1.7x** of ZFP on Hurricane

Empirical Evaluation – Rate-Distortion

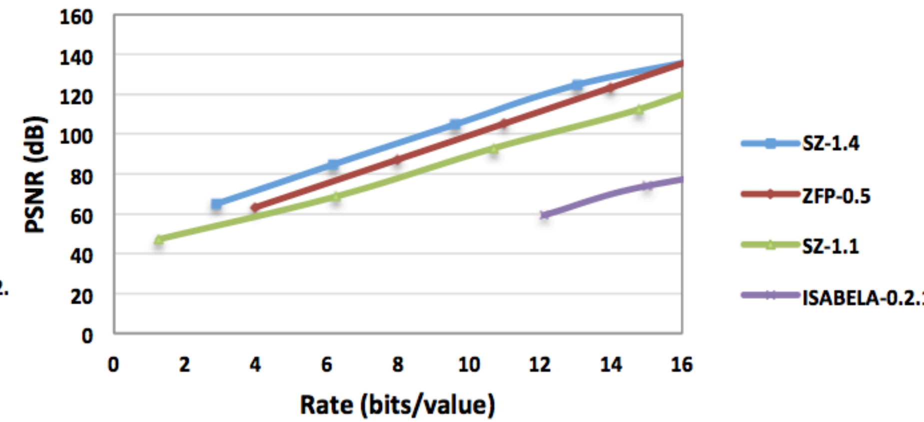
25% ~ ZFP (NRMSE)

ATM



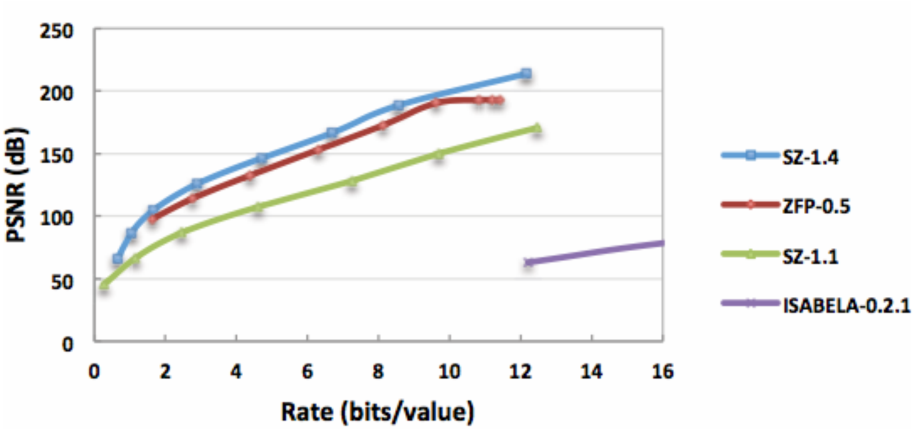
(a)

APS



(b)

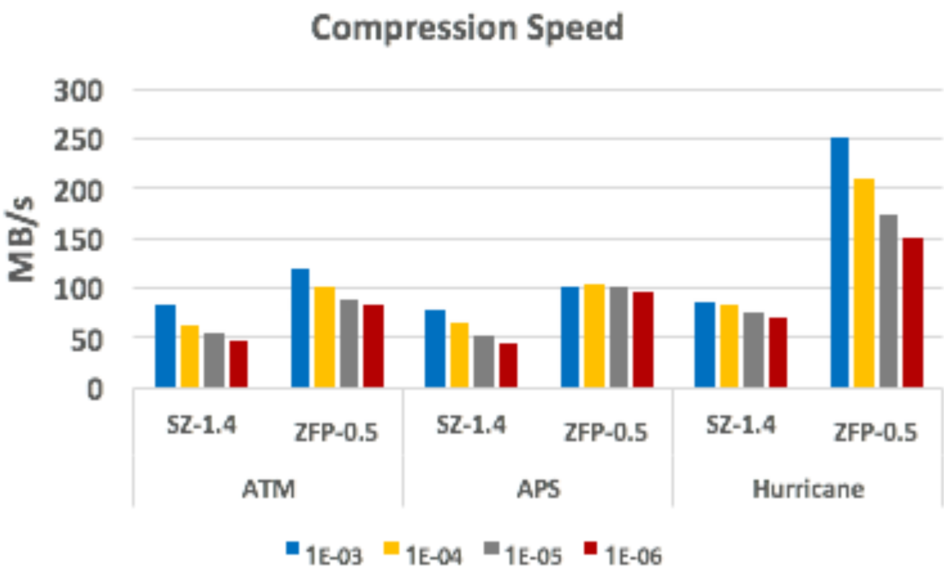
Hurricane

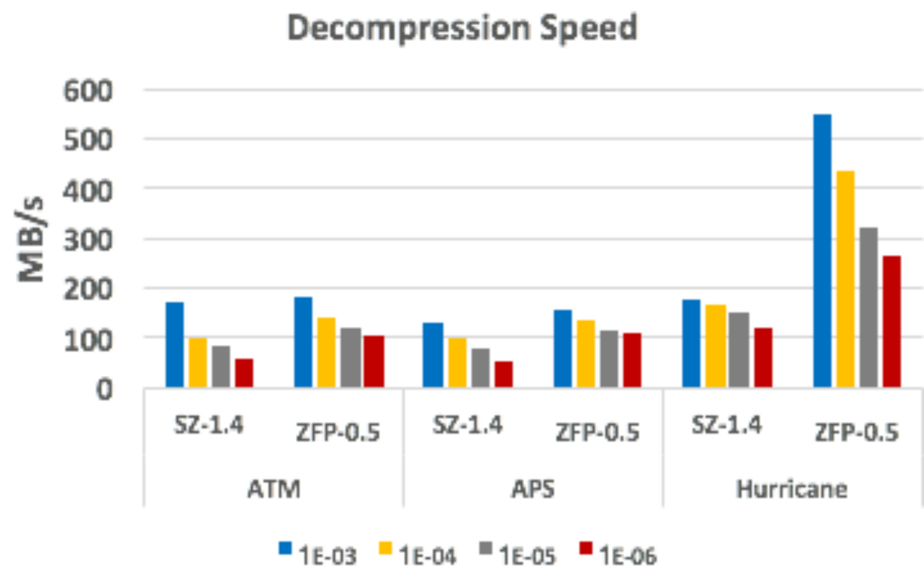


(c)

- ZFP: Best mode “fixed-accuracy“
- e.g., bit-rate = 8 bits/value (CR = 4)
  - 14 dB higher than ZFP on ATM
  - 9 dB higher than ZFP on APS
  - 11 dB higher than ZFP on Hurricane
- NRMSE: **25%** ~ ZFP on average

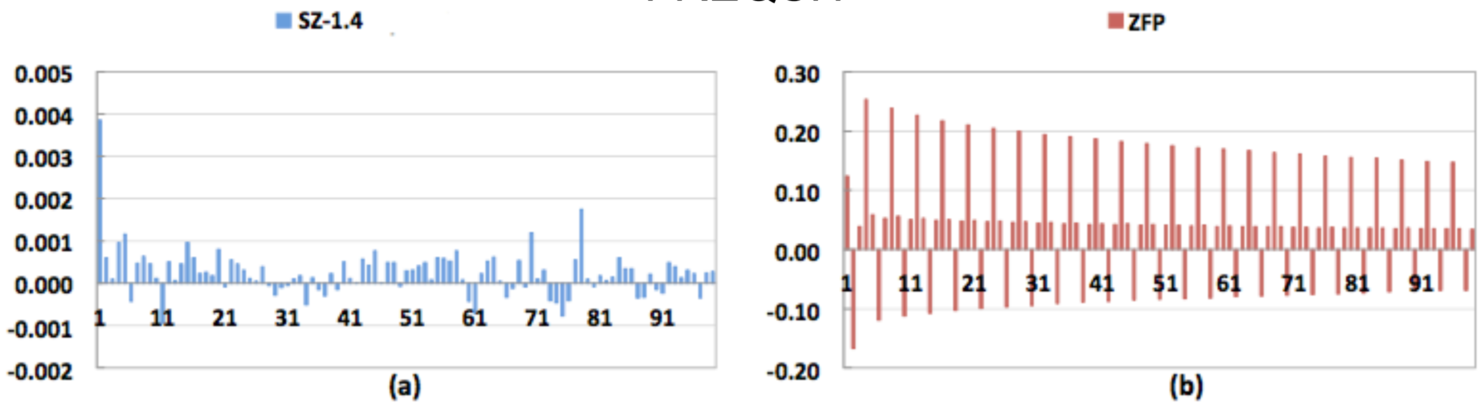
Empirical Evaluation – Comp/Decomp Speed



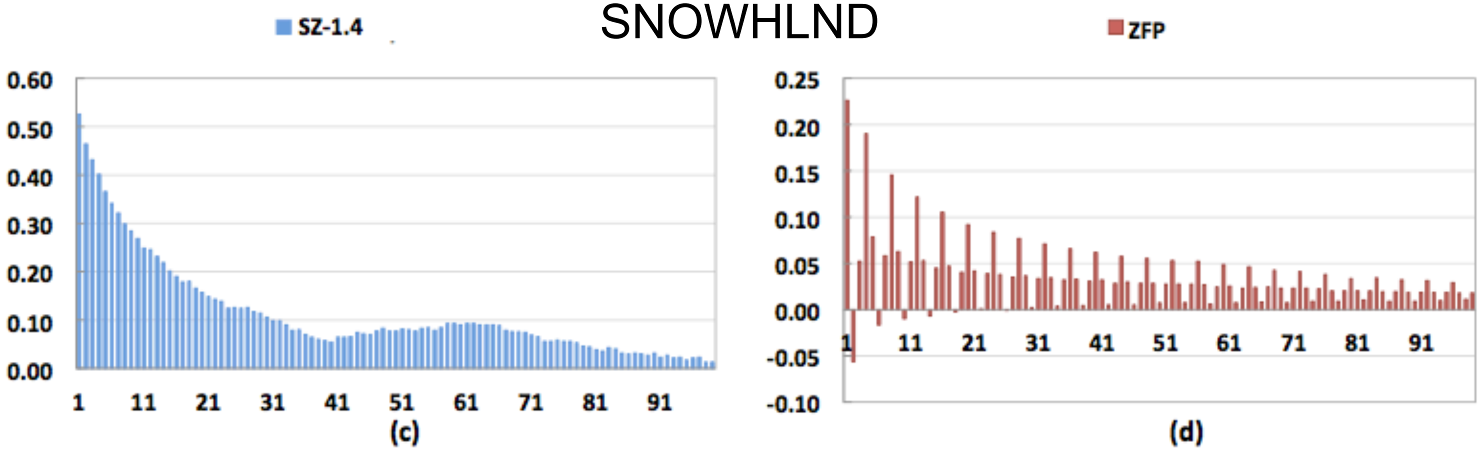


Empirical Evaluation – Autocorrelation of Errors

FREQSH



SNOWHLND



Empirical Evaluation – Parallel Compression (1)

- Parallel compression
  - In-situ: embedded in a parallel application
  - **Off-line**: MPI load data into multiple processes, run compression separately
- Experimental configurations
  - 2.6 TB ATM data sets with 11400 files
  - Blues cluster at ANL
  - Up to 1024 cores (64 nodes)
- **1 ~ 128 processes:** parallel efficiency stay **100%** - linear speedup
- **> 128 processes** (> 2 processes/node): parallel efficiency is decreased to **90%**
- This performance degradation is due to node internal limitations

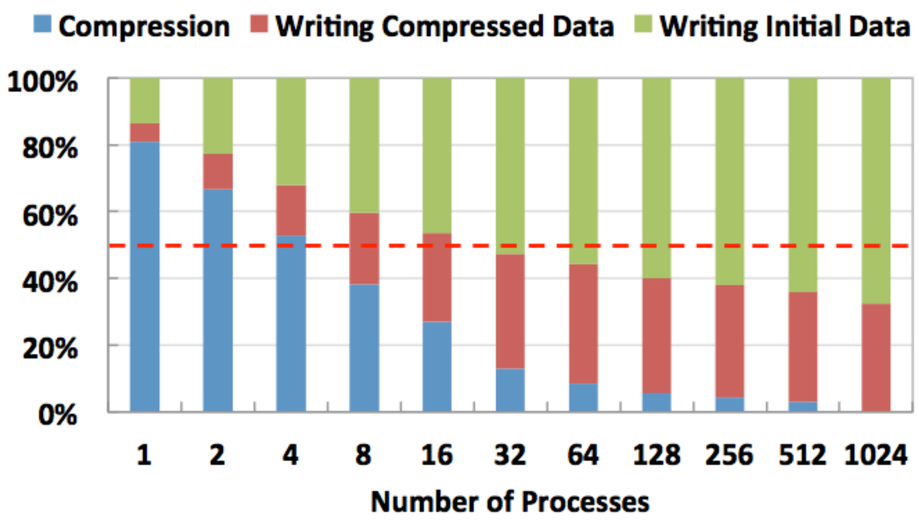
Table VII
STRONG SCALABILITY OF PARALLEL COMPRESSION USING SZ-1.4 WITH DIFFERENT NUMBER OF PROCESSES ON BLUES

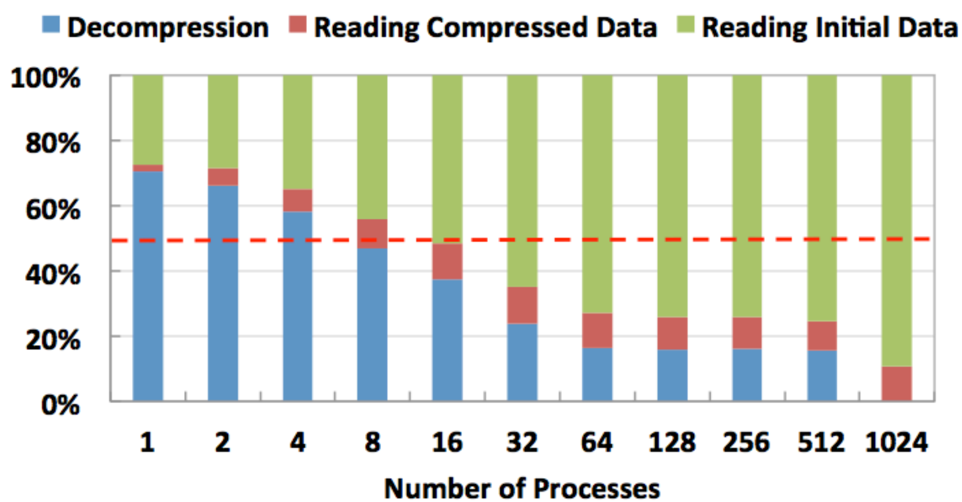
Number of Processes	Number of Nodes	Comp Speed (GB/s)	Speedup	Parallel Efficiency
1	1	0.09	1.00	100.0%
2	2	0.18	2.00	99.8%
4	4	0.35	3.99	99.9%
8	8	0.70	7.99	99.8%
16	16	1.40	15.98	99.9%
32	32	2.79	31.91	99.7%
64	64	5.60	63.97	99.9%
128	64	11.2	127.6	99.7%
256	64	21.5	245.8	96.0%
512	64	40.5	463.0	90.4%
1024	64	81.3	930.7	90.9%

Table VIII
STRONG SCALABILITY OF PARALLEL DECOMPRESSION USING SZ-1.4 WITH DIFFERENT NUMBER OF PROCESSES ON BLUES

Number of Processes	Number of Nodes	Decomp Speed (GB/s)	Speedup	Parallel Efficiency
1	1	0.20	1.00	100.0%
2	2	0.40	1.99	99.6%
4	4	0.80	4.00	99.9%
8	8	1.60	7.94	99.2%
16	16	3.20	16.00	99.9%
32	32	6.40	31.91	99.7%
64	64	12.8	64.00	99.9%
128	64	25.6	127.7	99.7%
256	64	49.0	244.5	95.5%
512	64	92.5	461.4	90.1%
1024	64	187.0	932.7	91.1%

Empirical Evaluation – Parallel Compression (2)





Number of Processes / Nodes > 32:

Time (writing compressed data + compression) < ***Time (***writing initial data***)***

Time (reading compressed data + decompression***)*** < ***Time (***reading inital data***)***

Outline

- Introduction
  - Large amount of scientific data
  - Limitations of lossless compression
- Existing lossy compressors and limitations
- Our Designs
  - Multidimensional / Multilayer Prediction Model
  - Adaptive Error-Controlled Quantization
- Metrics and Measurements
- Empirical Evaluation
  - Compression performance & Parallel evaluation
- Conclusion

Conclusions

- We derive a generic model for the multidimensional prediction to further use data's multidimensional information
- We propose an adaptive error-controlled quantization to deal with irregular and spiky data
- Our designs improve prediction-hitting rate significantly
- Compression ratio, rate-distortion better than second-best solution
- Save large amount of I/O time in parallel
- Furture work
  - Optimize SZ code to accelerate speed, especailly on high dimensional datasets
  - Develope SZ compressor for different architectures
  - Further reduce autocorrelation of compression errors

Thank you !

Welcome to use our SZ lossy compressor!

https://github.com/disheng222/SZ

Any questions are welcome!

Contact:

Dingwen Tao (dtao001@cs.ucr.edu)

Sheng Di (sdi1@anl.gov)

Acknowledgement: DOE EZ Project