HPDC 2024

Pisa, Italy

CereSZ: Enabling and Scaling Error-bounded Lossy Compression on Cerebras CS-2

Shihui Song, Yafan Huang, Peng Jiang, Xiaodong Yu, Weijian Zheng, Sheng Di*, Qinglei Cao, Yunhe Feng, Zhen Xie, and Franck Cappello

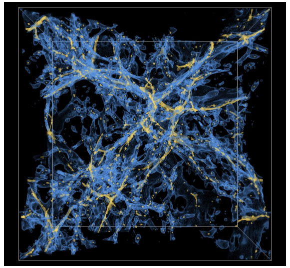
Presenter: **Robert Underwood**

The University of Iowa
Argonne National Laboratory
Stevens Institute of Technology
Saint Louis University
University of North Texas
Binghamton University, State University of New York

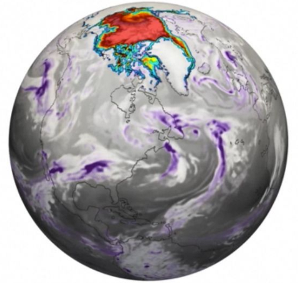
Lossy Compression in AI and HPC

- **Lossy compression** can reduce data size drastically.
  - ***Much higher compression ratio*** than lossless compression (limited to 2:1).
  - Introduced errors are controlled within a certain bound – ***error-bounded***.
- **Error-bounded lossy compression** is used by various domains.

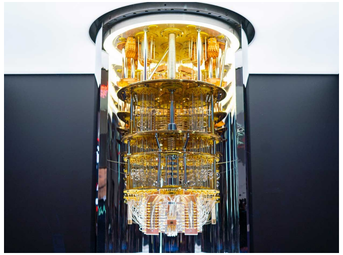
Deep Learning[1]
(e.g. Natural Language Processing)



Cosmology Simulation[2]
(e.g. NYX)



Climate Simulation[3]
(e.g. CESM-ATM)



Quantum Circuit Simulation[4]
(e.g. Grover)

1. [**TelecomReview'2020**] The rise of emotionally intelligent artificial intelligence
2. [**News@CMU'2021**] Machine Learning Accelerates Cosmological Simulations
3. [**TechReview@MIT'2018**] What the hell is a climate model—and why does it matter?
4. **[IEEESpectrum'2020]** IBM's concept of quantum volume tries to measure quantum computing progress in ways beyond counting qubits

An Emerging AI Chip System: Cerebras CS-2

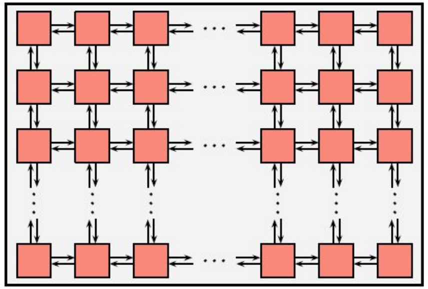
- Cerebras is critical to accelerate many **scientific computing applications**.
  - 3D Fast Fourier Transform: ***959 ms*** for ***512x512x512*** complex input array[1].
  - Matrix-Vector Multiplications: ***92.58 PB/s*** on ***35,784,000*** processing elements[2].
- However, Cerebras CS-2 system is processing **massive data**.
  - Large Language Model Training: GPT-3 had ***175 billion*** parameters, and this number is increasing to ***1 trillion*** in near future models such as MSFT-1T[3].
  - Seismic Imaging: ***1.8 TB*** data for a 4.5 seconds and 45 Hz flat wave[2].

Performing efficient data reduction within Cerebras CS-2 system is necessary!

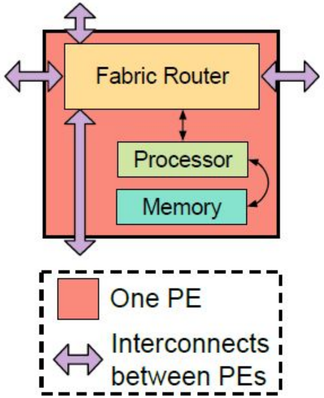
1. **[ICS'2023]** Wafer-Scale Fast Fourier Transforms
2. **[SC'2023]** Scaling the "Memory Wall" for Multi-dimensional Seismic Processing with Algebraic Compression on Cerebras CS-2 Systems.
3. **[IEEE Micro'2023]** Cerebras Architecture Deep Dive: First Look Inside the Hardware/Software Co-Design for Deep Learning

Background: Cerebras CS-2 System

- **Dataflow Architecture:** **Wafer-scale engine (WSE)** is the central processor of Cerebras CS-2 system.

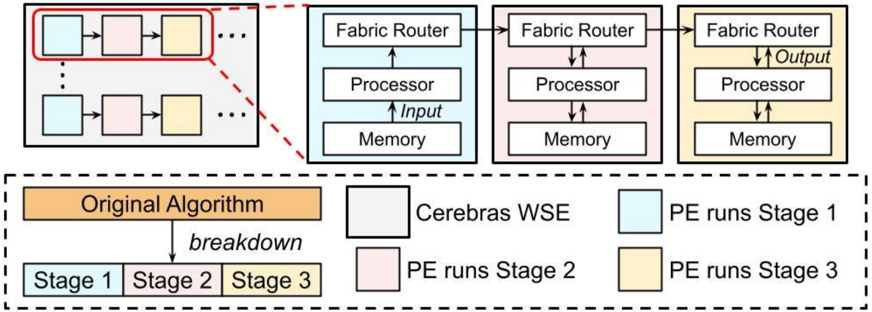


Cerebras WSE, where each node denotes one PE



Structure of a single PE

- **Parallel Processing**: Map computing stages into consecutive PEs in the same row.

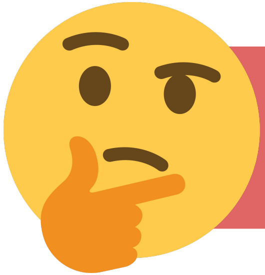


Pipeline parallelization on Cerebras WSE

- **Implementation**: A language called CSL provided by Cerebras.

Challenges: Deploying Compression on Cerebras

- **Cerebras Feature 1: Unconventional Memory Structure**
  - No traditional global memory as NVIDIA GPUs.
  - Local memory for each PE is limited (up to 48 kB).
- **Cerebras Feature 2: Spatial Constraints from Data-Flow Design**
  - Each PE can only access data from its neighbors.
  - The data movement directions could influence the performance.

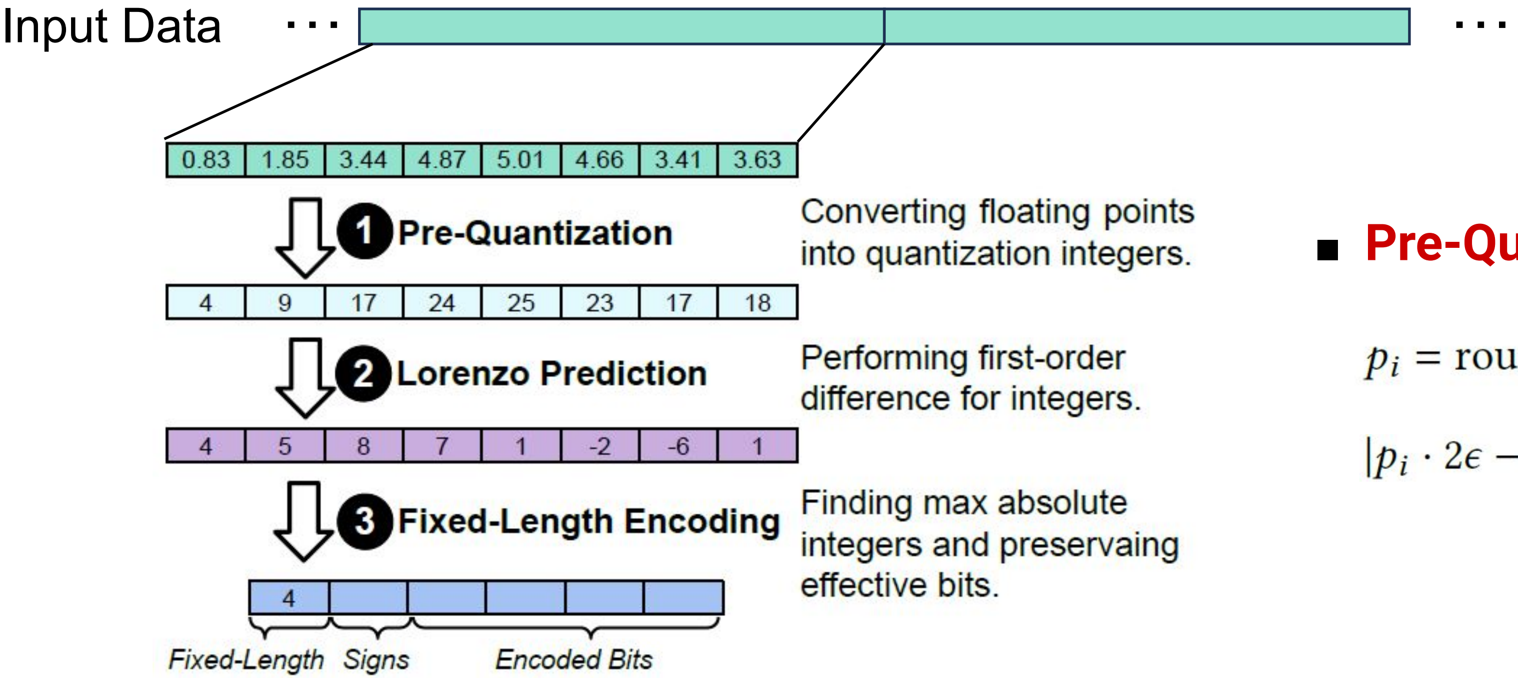


How to perform lossy data compression that efficiently utilizes Cerebras CS-2 architectural characteristics?

1. [**Cerebras White Paper**] Cerebras Systems: Achieving Industry Best AI Performance Through A Systems Approach

CereSZ: Compression Algorithm

- **An overview of the compression algorithm**



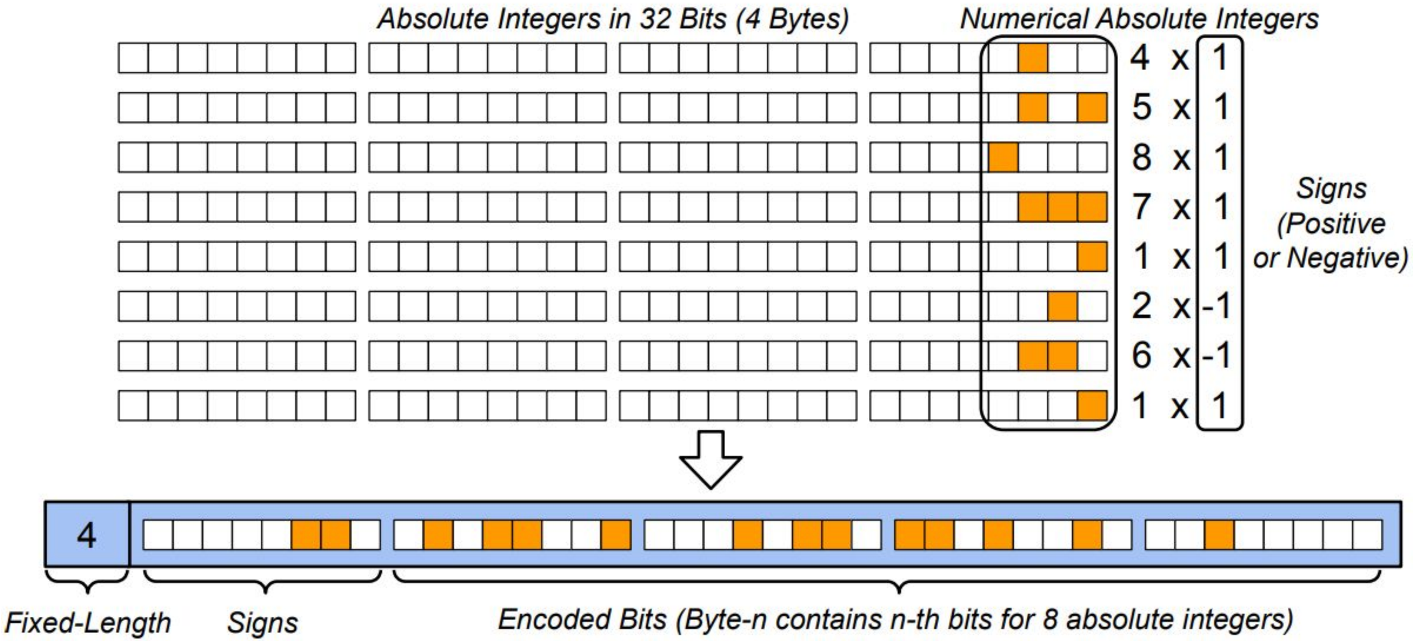
- **Pre-Quantization**

$$p_i = \text{round}\left(\frac{e_i}{2\epsilon}\right)$$

$$|p_i \cdot 2\epsilon - e_i| \leq \epsilon$$

CereSZ: Compression Algorithm

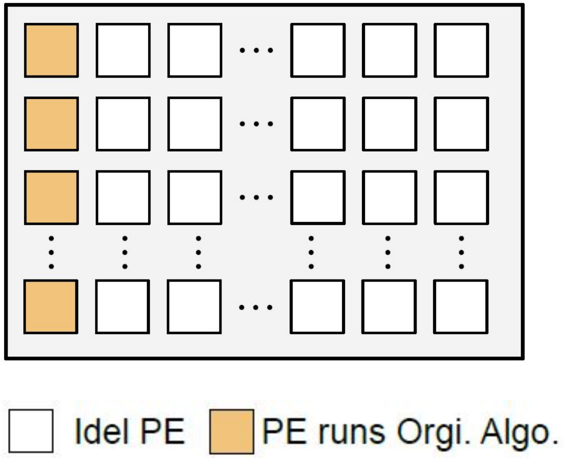
- **Fixed-length Encoding**



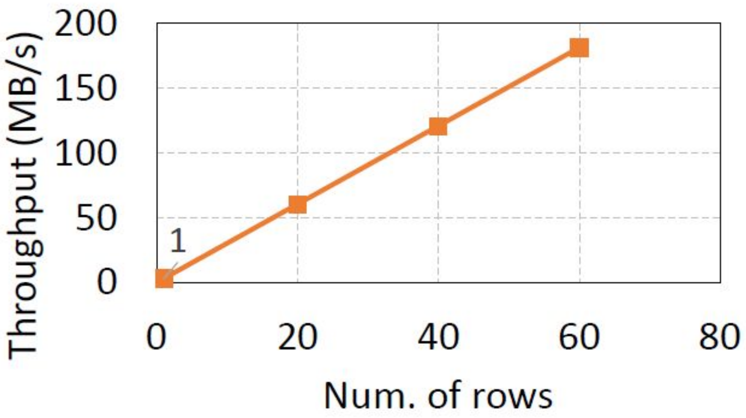
Compression Ratio:
8*4/(1+1+4)=5.33

Perf. Opt. 1: Data Parallelism (Block-level)

- Data Parallelism with Data Blocking



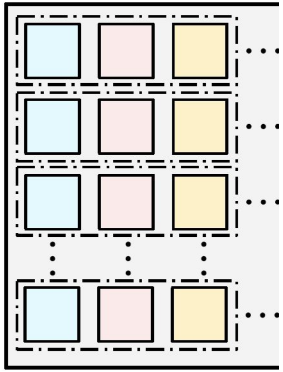
- Throughput with different numbers of PE rows



Throughput = Input Data Size / compression time

Perf. Opt. 2: Pipeline Parallelism (Stage-level)

- Pipeline Parallelism for 3 Stages



Idel PE

PE runs Stage 1 PE runs Stage 3

Algorithm 1: Evenly distributing n sub-stages across m PEs

Input: The stages: $s_1, s_2, ..., s_n$; Total cycles of all stages: C; Number of PEs: m;

Output: Stage group assigned to PEs: $G_1, G_2, ..., G_m$

1 Initialize $G_1 = \{\}, G_2 = \{\}, ..., G_m = \{\}$

2 **for** *each stage group* G_i *in* $\{G_1, G_2, ..., G_{m-1}\}$ **do**

3 **while** *The sum of runtime of the stages in* $G_j < \frac{C}{m}$ **do**

4 move the next s_j to G_i

5 $G_m = \{s_1, s_2, ..., s_n\} - (G1 \cup G2 \cup ... \cup G_{m-1})$

- Breakdown cycles for compression

Encd.
37124
29181
27188

Addition
1033
1038
1049

tLength	Bit-shuffle
1386	33609
1370	25675
1385	23694

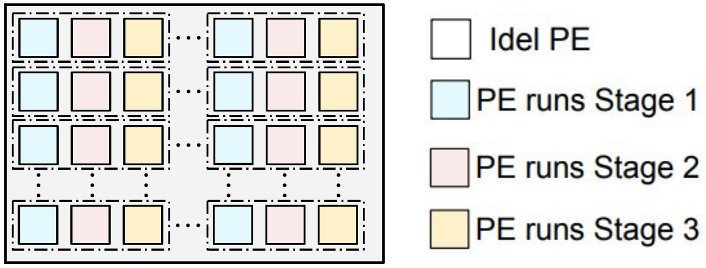
Fixed-Length for CESM-ATM, HACC, and QMCPack: 17, 13, and 12

Bit-shuffle -> several 1-bit shuffle

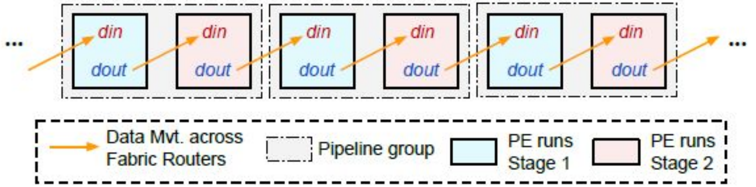
Stages: Multiplication, Addiction, Lorenzo Prediction, Sign, Max, GetLength, 1-bit shuffle

Perf. Opt. 3: Data Parallelism (Pipeline-level)

- Data Parallelism for Multiple Pipelines



- Passing data with 2-length pipeline



- Pseudocode runs on the first PE of each pipeline

```
task relay() void {
    // Receive the input dsd and activate computeColor once the
    // current PE receives it own data block
    if (nblock == (total_cols-cur_col)/pipeline_length)){
        @mov32(data, din, .{.async = true, .activate =
        computeColor}); nblocks = 0;}}
    // Pass the data blocks for right PEs and activate relayColor again
    else{
        @mov32(dout, din, .{.async = true});
        nblocks += 1;
        @activate(relayColor);}

task compute() void {
    // Activate relayColor to run relay task again
    @activate(relayColor);}
    // Execute substages assigned to the PE
    // Send results to next PE in the pipeline}

// Bind two colors to their corresponding tasks
@bind_task(relay, relayColor);
@bind_task(compute, computeColor);
```

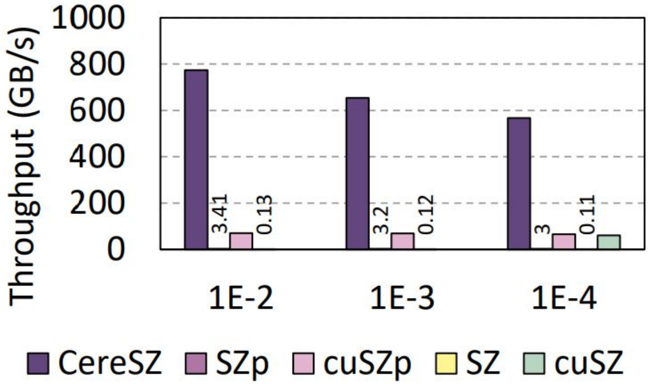
Evaluation: Settings

- **Neocortex Cerebras CS-2**
  - 512×512 processing elements (PEs)
  - 850MHz clock frequency
- **Baseline Compressor**
  - *SZ*[1] : AMD EPYC 7742 CPUs
  - *cuSZp*[2] : NVIDIA A100 GPU (40GB)
  - *SZp*[2] : AMD EPYC 7742 CPUs
  - *cuSZ*[3] : NVIDIA A100 GPU (40GB)
- **Evaluation Metrics**
  - Throughput (GB/s)
  - Compression ratio
  - Reconstructed data quality
- **HPC Datasets**
  - *Hurricane*: weather simulation
  - *NYX*: cosmology simulation
  - *QMCPack*: quantum computing
  - *RTM*: seismic imaging
  - *HACC*: cosmic simulation
  - *CESM-ATM*: climate simulation

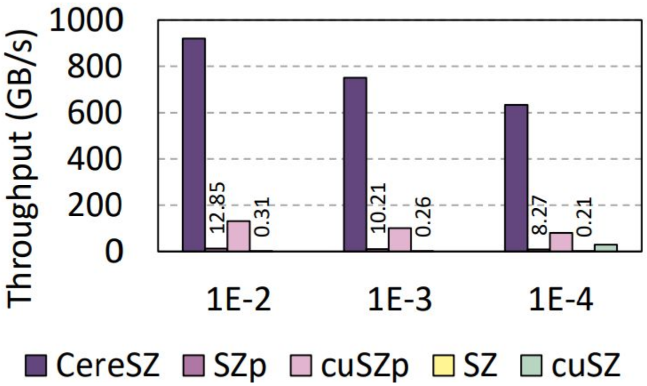
1. [**IEEE TBD'2022**] SZ3: A Modular Framework for Composing Prediction-Based Error-Bounded Lossy Compressors
2. [**SC'2023**] cuSZp: An Ultra-fast GPU Error-bounded Lossy Compressor with Optimized End-to-End Performance
3. [**PACT'2020**] cuSZ: An Efficient GPU-Based Error-Bounded Lossy Compression Framework for Scientific Data

Evaluation: Throughput

- Compression and decomrpession throughput on RTM dataset



Compression



Decompression

- On average, CereSZ can achieve **457.35 GB/s** and **581.31 GB/s** for compression and decompression throughput, which is **4.9** and **4.8** times faster compared with cuSZp.

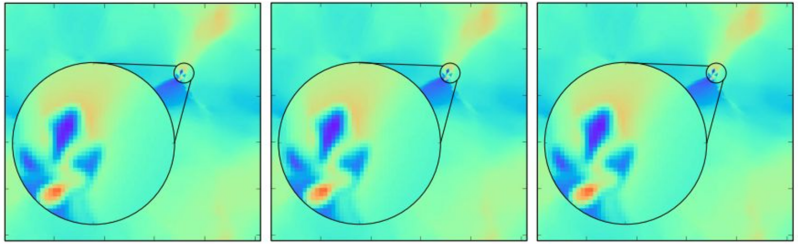
Evaluation: Compression Ratio

	REL	CESM-ATM range	CESM-ATM avg	HACC range	HACC avg	Hurricane range	Hurricane avg	NYX range	NYX avg	QMCPack range	QMCPack avg	RTM range	RTM avg
CereSZ	1E-2	2.67~21.60	8.73	4.66~9.18	6.82	5.21~28.82	17.10	7.83~31.98	20.22	9.59~19.67	14.63	10.52~31.99	23.46
	1E-3	2.13~16.10	6.49	3.18~4.91	4.05	3.41~24.37	12.57	4.54~31.84	14.05	5.31~9.02	7.16	5.94~31.98	17.73
	1E-4	1.68~13.42	5.11	2.38~3.20	2.83	2.53~19.71	9.64	3.10~29.74	9.61	3.48~4.97	4.23	3.79~31.96	12.87
SZp	1E-2	9.91~70.48	23.72	10.16~13.62	11.56	10.80~88.94	40.26	12.21~127.80	67.58	12.44~22.45	17.45	14.22~127.94	67.51
	1E-3	6.70~69.15	20.14	3.82~9.63	5.39	7.44~57.42	23.92	8.62~125.55	40.16	6.08~11.60	8.84	7.91~127.79	43.40
	1E-4	4.22~67.65	17.03	3.49~5.51	3.57	4.49~37.08	15.29	4.91~98.23	23.41	3.79~6.57	5.18	4.73~127.51	28.19
cuSZp	1E-2	2.84~43.75	12.56	5.24~10.08	7.63	5.94~88.88	38.70	9.60~127.80	66.73	12.44~22.21	17.33	13.97~127.95	66.97
	1E-3	2.25~25.86	8.46	3.43~5.20	4.31	3.71~56.88	22.31	5.09~125.55	38.44	6.08~10.08	8.08	6.90~127.80	42.29
	1E-4	1.75~19.59	6.24	2.53~3.39	2.96	2.70~36.66	14.36	3.35~98.23	22.14	3.79~5.56	4.68	4.17~127.52	27.43
SZ	1E-2	26.13~4.0E+4	2.2E+3	16.58~931.76	217.94	23.76~404.71	110.33	1.3E+3~1.2E+5	2.3E+4	17.10~727.13	372.11	23.57~1.3E+5	4.4E+3
	1E-3	9.30~2.9E+4	941.39	6.11~30.97	15.57	8.81~105.49	35.67	84.55~1.8E+4	3.2E+3	6.37~221.11	113.74	9.27~2.3E+4	894.69
	1E-4	5.04~2.9E+4	825.49	3.74~8.92	5.75	4.63~48.46	18.72	14.38~2.6E+3	471.61	3.88~66.09	34.99	5.30~1.6E+4	548.91
cuSZ	1E-2	19.18~25.33	22.89	N/A	N/A	15.35~28.62	22.53	28.71~31.57	30.22	7.50~21.55	14.53	N/A	N/A
	1E-3	11.34~25.16	18.48	N/A	N/A	8.91~23.61	15.97	N/A	N/A	4.26~17.70	10.98	N/A	N/A
	1E-4	5.38~24.43	12.47	N/A	N/A	3.37~17.25	8.36	10.75~31.28	16.22	N/A	N/A	3.67~30.84	11.63

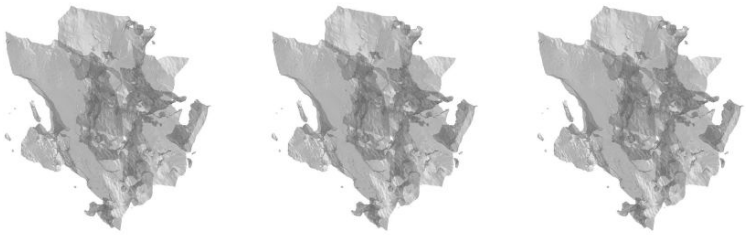
- CereSZ has lower compression ratios than CPU-based compressor SZ
- But CereSZ is inline with GPU-based alternatives such as cuSZ and cuSZp.

Evaluation: Data Quality

- Slice and Isosurface visualization analysis



(a) Ori. Slice (b) cuSZp Slice (c) CERESZ Slice



(d) Ori. Isosurface (e) cuSZp Isosurface (f) CERESZ Isosurface

- CereSZ has **the same** data quality as cuSZp

Summary

- CereSZ is **the first** error-bounded lossy compressor on Cerebras CS-2 system.
- **457.35 GB/s and 581.31 GB/s** for compression/decompression throughput.
- CereSZ has linear speedups across the rows and columns of the 2D mesh of computing units (i.e. PEs) on Cerebras.
- CereSZ also achieves **similar compression ratio** with GPU alternatives.
- CereSZ demonstrates potential to preserve high data quality.

Shihui Song
University of Iowa
shihui-song@uiowa.edu
https://songshsongsh.github.io