cuSZ(+): Optimizing Error-Bounded Lossy Compression for Scientific Data on Modern GPUs

Washington State University

Jiannan Tian (WSU)
Dr. Franck Cappello (ANL)
Dr. Sheng Di (ANL)
Dr. Dingwen Tao (WSU)

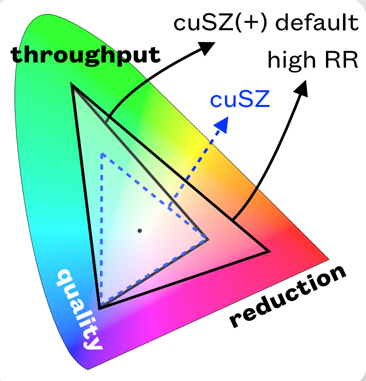
Argonne National Laboratory

Big-Data Scientific Application

application	data scale	to reduce
HACC cosmology simulation	**20 PB** per one-trillion-particle simulation	**10x** in need
CESM climate simulation	20% vs **50%** of h/w budget for storage 2013 vs 2017	**10x** in need
APS-U High-Energy X-Ray Beams Experiments	hundreds of **PB** brain initiatives	**100x** in need

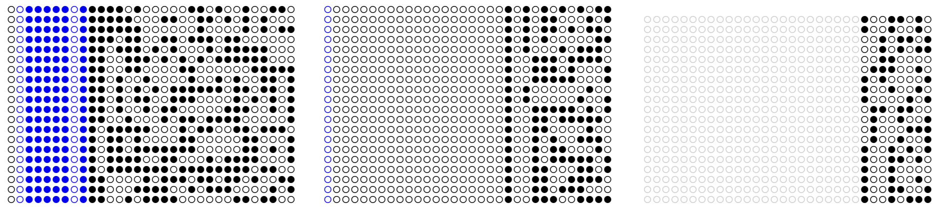
Data management is a real-world problem to address when we advance in scientific exploration.

Data Reduction



+ **high-quality** data-reconstruction for accurate post-analysis
+ data **reduction rate** at 10x in need
+ high-**throughput** processing capability to ease I/O and communication pressure

CESM-CLDHGH, data range: 0.89401052, error-bound: 1e-4, relative to the range



float (32 effective bits)
mantissa randomness

int (13 effective bits, incl. signum)
rounding, type-casting
integerized regarding error bound

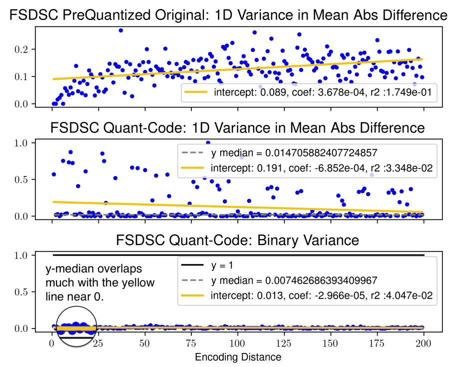
unsigned int (8 effective bits)
prediction → recorded errors
(of prediction from the previous)

To lower bit randomness: prediction-based SZ.

SZ[1,2] Lossy Compression Essence

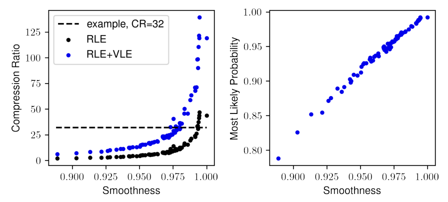
+ prediction
+ error-control
+ compress error-control representation integer

"Smoothness" for RLE



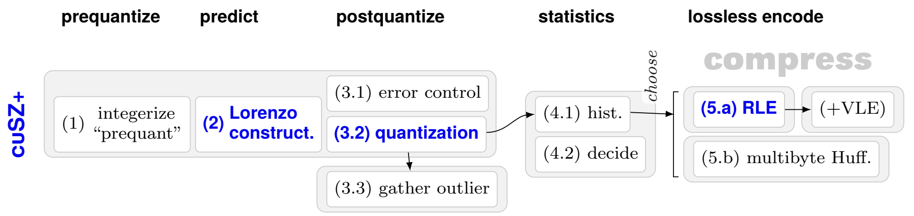
$$2\gamma(s_1, s_2) = \mathrm{var}(s_1, s_2) = E\left[|Z(s_1) - Z(s_2)|\right]$$

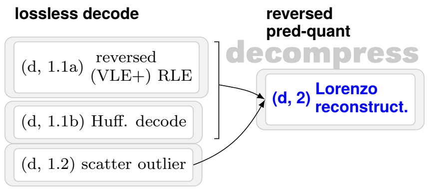
Madogram (variogram variant) to determine "smoothness"



"smoothies" <=> rate <=> histogram
e.g., entropy < 1.09 bit, use RLE

From cuSZ[3] to cuSZ(+)





+ cuSZ[3], the GPU adoption of SZ[1,2]
  ! no **pattern finding**
  ! reduction limited to 32x
+ fully parallelized compress-time
  ! not in **decompress**-time
+ cuSZ(+) addresses the problems.

Fine-Grained Reconstruction

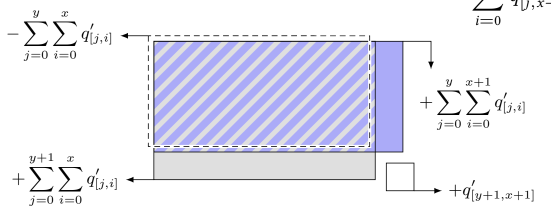
2D Lorenzo predictor[2] $p_{[y,x]} = -d_{[y-1,x-1]} + d_{[y-1,x]} + d_{[y,x-1]}$

local reconstruction $d_{[y,x]} = p_{[y,x]} + q_{[y,x]}$ **general term** $\sum_{j=0}^{y}\sum_{i=0}^{x} q_{[j,i]}$

proof by induction

$$d^{\bullet}_{[y+1,x+1]} = -\sum_{j=0}^{y}\sum_{i=0}^{x} q'_{[j,i]} + \sum_{j=0}^{y}\sum_{i=0}^{x+1} q'_{[j,i]} + \sum_{j=0}^{y+1}\sum_{i=0}^{x} q'_{[j,i]} + q'_{[y+1,x+1]}$$

$$= \sum_{i=0}^{y} q'_{[j,x+1]} + q'_{[y+1,x+1]} + \sum_{j=0}^{y+1}\sum_{i=0}^{x} q'_{[j,i]} = \sum_{j=0}^{y+1}\sum_{i=0}^{x+1} q'_{[j,i]}.$$



With internal cancellation, it is computationally cheap and done by *N*-D partial-sum.

General optimization regarding data path: (1) coalescing load to shared memory from global memory, (2) coarsening by assigning multiple items to one thread, (3) in-warp shuffle, (4) coalescing access to shared memory for out-of-warp data exchange, (5) coalescing store to global memory. (`NVIDIA::cub` is only used for 1D case.)

Evaluation Setup

Platform
A100, ALCF-ThetaGPU
V100, TACC-Longhorn

Datasets
1D HACC, 2D CESM, 3D Hurricane, Nyx, QMC

Specification
+ V100: 900 GiB/s, 14 TFLOPS
+ A100: 1555 GiB/s (1.7x), 19 TFLOPS (1.4x)

Evaluation: Reduction Rate

With intrinsic high compressibility, RLE is greater than VLE, (RLE+VLE) is even greater.

	cuSZ	ours		ours	
	VLE	RLE	gain	RLE+VLE	gain
FSDSC	23.88	26.10	1.09×	71.35	2.99×
FSDTOA	26.10	43.65	1.67×	119.17	4.57×
ODV_bcar1	25.83	37.28	1.44×	110.51	4.28×
ODV_bcar2	25.83	30.71	1.19×	89.98	3.48×
ODV_dust1	26.10	22.91	-	67.72	2.59×
ODV_dust2	26.37	24.02	-	70.98	2.69×
ODV_dust3	26.10	33.29	1.28×	98.22	3.76×
ODV_dust4	26.10	46.81	1.79×	139.27	5.34×
ODV_ocar1	24.11	41.17	1.71×	121.59	5.04×
ODV_ocar2	24.11	33.79	1.40×	98.63	4.09×
PRECSC	25.83	19.50	-	58.92	2.28×
SNOWHLND	25.57	21.18	-	63.33	2.48×
SOLIN	26.10	43.65	1.67×	119.17	4.57×

+ CESM, error bound at 1e-2, relative to range
+ Considering the throughput, RLE only can perform well enough on some data fields.
+ can also append VLE for higher reduction rate

Reference

[1] S. Di and F. Cappello, "Fast error-bounded lossy HPC data compression with SZ," in *2016 IEEE International Parallel and Distributed Processing Symposium*, Chicago, IL, USA: IEEE, 2016, pp. 730–739.

[2] D. Tao, S. Di, Z. Chen, and F. Cappello, "Significantly improving lossy compression for scientific data sets based on multidimensional prediction and error-controlled quantization," in *2017 IEEE International Parallel and Distributed Processing Symposium*, Orlando, FL, USA: IEEE, 2017, pp. 1129–1139.

[3] J. Tian *et al.*, "cuSZ: An efficient gpu-based error-bounded lossy compression framework for scientific data," in *Proceedings of the ACM International Conference on Parallel Architectures and Compilation Techniques*, 2020, pp. 3–15.

Evaluation: Throughput

cuSZ[3] to cuSZ(+) on V100

V100		HACC	CESM	Hurr	Nyx	QMC
Lorenzo construct	cuSZ	207.7	252.1	175.8	200.2	189.6
	cuSZ(+)	307.4	273.9	229.9	296	298.6
		1.48×	1.09×	1.31×	1.48×	1.57×
Huffman encode	cuSZ	54.1	57.2	55.2	58.8	61
	cuSZ(+)	58.3	107.7	111.2	120.5	110.8
		1.08×	1.88×	2.01×	2.05×	1.82×
Lorenzo reconstruct	cuSZ	16.8	58.5	43.9	29.7	22.4
	cuSZ(+)	313.1	254.2	218.4	238.1	255.5
		18.64×	4.35×	4.97×	8.02×	11.41×

Lorenzo construction kernels:
+ **1.48×** for 1D, **1.09×** for 2D, and **1.45×** for 3D
+ The lowest: 175.8 GB/s to 229.9 GB/s, **+30.7%**

scale cuSZ(+) to A100 from V100

size in MB		1071.8	24.7	95.4	512.0	601.5
		HACC	CESM	Hurr	Nyx	QMC
Lorenzo construct	V100	328.3	273.9	199.0	296.0	298.6
	A100	501.1	466.8	429.0	481.3	492.9
		1.53×	1.70×	2.16×	1.63×	1.65×
Huffman encode	V100	58.3	107.7	111.2	120.5	110.8
	A100	174.6	121.6	206.0	217.2	198.4
		2.99×	1.13×	1.85×	1.80×	1.79×
Lorenzo reconstruct	V100	308.7	267.0	200.1	251.7	255.5
	A100	504.4	495.3	345.5	398.6	384.0
		1.63×	1.86×	1.73×	1.58×	1.50×

+ Lorenzo construction: 1.53× to 2.16×
+ Huffman encoding: 1.13× to 2.99×
+ Lorenzo reconstruction: 1.50× to 1.86×
+ Faster HBM2e helps a lot.

Acknowledgement

This R&D was supported by the Exascale Computing Project (ECP), Project Number: 17-SC-20-SC, a collaborative effort of two DOE organizations—the Office of Science and the National Nuclear Security Administration, responsible for the planning and preparation of a capable exascale ecosystem. This repository was based upon work supported by the U.S. Department of Energy, Office of Science, under contract DE-AC02-06CH11357, and also supported by the National Science Foundation under Grants SHF-1617488, SHF-1619253, OAC-2003709, OAC-1948447/2034169, and OAC-2003624.