SZ Lossy Compression for Scientific Datasets

Sheng Di, Franck Cappello
sdi1@anl.gov, cappello@mcs.anl.gov

MCS, Argonne National Laboratory

**ECP EXASKY/NYX:**
Cosmology simulation
X velocity field

Original

**SZ Compression ratio:** 143
PSNR=64dB

**Signal/error**
$\sim 10^3$
ANL SZ Framework Design Principles

- Error bounded (point-wise, PSNR)
- Multi-stages, Multi-algorithms, Multi-strategies (space, time)
- Prediction based: allowing customization of predictor
- Block based to allow for random access decompression

Current version: **SZ 2.1.11**
(Previous versions: SZ 1.1, SZ 1.4)
SZ Applications

- Integrated into multiple scientific applications
- Evaluated/used by 20+ institutes/universities.
- 2000+ downloads/year
- Integrated in Spack.
- 40+ papers documenting each progress, optimization, application

SZ compressor: szcompressor.org
SZ Use-cases

- Reducing I/O time for HPC simulations
  - ECP apps: HACC, NWCHEMeX
- Reducing storage footprint
  - ECP apps: HACC, NYX
- Reducing data stream intensity
  - ECP apps: EXAFEL
- Reducing memory footprint
  - ECP apps: GAMESS
- Reducing Communication cost
  - ECP apps: GAMESS
- Eliminating recomputation cost
  - ECP apps: GAMESS

Publication: IJHPCA2017
SZ Software Ecosystem

Compression Benchmark (SDRBench, Squash, etc.)

Compression Assessment Tool/Library (Z-checker, LANL VizAly-Foresight, libpressios, etc.)

Scientific Data Compressors
- Generic Lossy compressors (SZ, ZFP, ...)
- Customized Lossy Compressors (Pastri, DeepSZ, etc.)
- Lossless Compressor (FPZIP, FPC, Zstd, etc.)

Supporting various platforms/environments
- Support multi-CPU-cores
- Support GPU
- Support FPGA
- Support I/O libraries (HDF5, ADIOS, NetCDF, GIO)

DOE Applications

Repository of reference scientific dataset for compression benchmarking

Standard tools to assess compression quality

Data compression strategies

Simplifying I/O and accelerating throughput

The items in blue are developed/focused in our team.
Success story: Cosmology simulation
ECP HACC


Particle dataset: 6 x 1D array (x, y, z, vx, vy, vz)
Very hard to compress

Preferred error controls:
• Point wise max error (Relative) bound
• Absolute (position), Relative (Velocity)

SZ 2.0: CR \( \sim 5 \) (~6bits/value) at \( 10^{-3} \) error bound
Success story: Quantum Chemistry

ECP GAMESS

- Two-Electron Integrals in Quantum Chemistry

The goal is to obtain the wavefunction of a chemical system by solving the Schrödinger equation.

We customized an efficient error-bounded compressor for GAMESS two-electron integrals dataset and successfully integrated it into GAMESS Fortran code.

CR: 17 at $10^{-11}$
1.5X end-to-end overall execution performance gain is observed.

Best paper award
at IEEE Cluster 2018
GPU performance

cuSZ: cuda based SZ, kSZ: kokkos based SZ

Key techniques: dual-quantization/prediction, Huffman on GPU

Compression Performance on GPU:

- Overall: HACC (A100): 66.7GB/s, NYX (A100): 66.9GB/s, QMCPack (A100): 62.6GB/s

- Fastest Huffman encoding implementation on GPU: 135GB/s~175GB/s on A100.


SZ as a community software

Core-group:
- (Lead) Argonne National Laboratory
  Dr. Franck Cappello, Dr. Sheng Di, Dr. Ali Murat Gok
- Washington State University
  Dr. Dingwen Tao, Jiannan Tian, Cody Rivera, Sian Jin, Chengming Zhang
- University of California, Riverside
  Kai Zhao, Sihuan Li, Jinyang Liu
- Clemson University
  Dr. Jon Calhoun, Robert Underwood
- Missouri University of Science and Technology
  Dr. Xin Liang

Other Contributors:
Dr. Martin Herbordt, Dr. Qingqing Xiong, Dr. Wen Xia, Xiangyu Zou
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.