SAND2018-13937C

Sandia National Laboratories

SNL ATDM: In-situ Compression with ParaView/TuckerMPI

Hemanth Kolla (PI), Ron Oldfield (PM)
Tom Otahal, Gavin Baker, Jeff Mauldin, Tammy Kolda, Ken Moreland

Overview

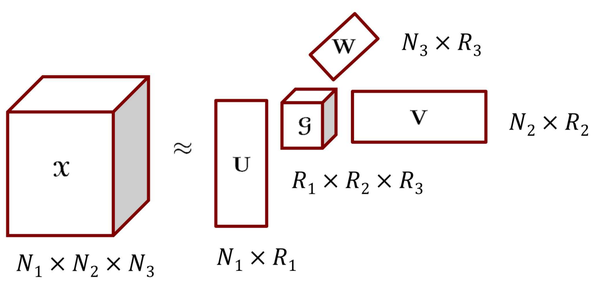
Motivation: Exascale apps are anticipated to generate volumes of data that will likely overwhelm capacity and throughput of data storage resources.

Objective: Deliver efficient, scalable, in-situ, orders-of-magnitude compression with user-controlled accuracy.

Solution:
- Higher-order tensor decomposition (Tucker) for orders-of-magnitude (lossy) compression.
- Integrate with ParaView/Catalyst for in-situ capability.

TuckerMPI

Tucker ≡ higher order Matrix SVD



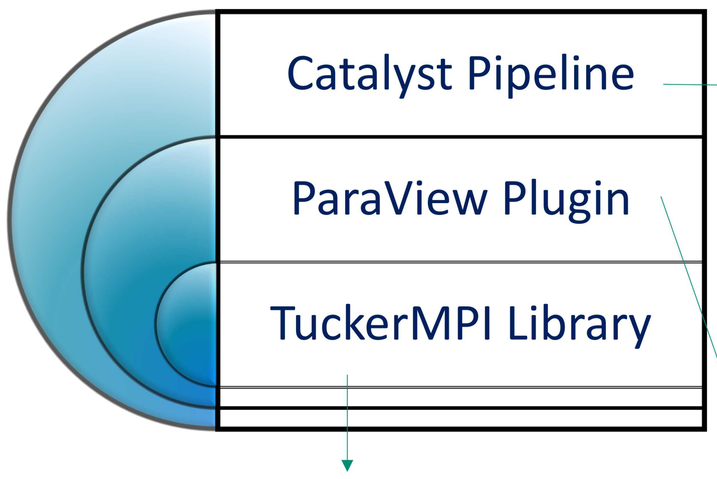
$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$$

$$\text{CR} \approx \frac{N_1 \times N_2 \times N_3}{R_1 \times R_2 \times R_3}$$

TuckerMPI [1]: Scalable parallel implementation

- MPI-based parallel implementation of "sequentially truncated higher order SVD".
- Tucker factorization, compression, of arbitrary order tensor distributed on a Cartesian topology.
- Three main kernels parallelized: Gram matrix, eigendecomposition, tensor-times-matrix.
- Interfaces with MPI-IO for reading input, writing output tensors.
- https://gitlab.com/tensors/TuckerMPI

ParaView Integration for In-Situ Capability



- Dynamic library interface to TuckerMPI functionality
- API for Tensor data structures, MPI domain decomposition maps/objects
- API for key kernels: tensor matricization, ST-HOSVD, Gram matrix computation.

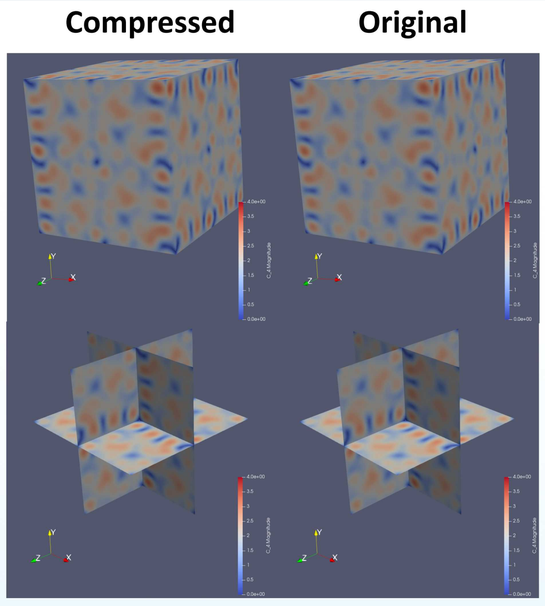
ParaView Catalyst [2]: Use-case library with adaptable API for *in-situ* analysis and/or visualization tasks.

```
def DoCoProcessing(datadescription):

    global coprocessor
    coprocessor.UpdateProducers(datadescription)

    LoadPlugin('/path/to/plugin/libSMTuckerMPICompression.so', remote=False, ns=globals())

    vtk_uniq_test_name_9vtm = coprocessor.CreateProducer(datadescription, 'input')

    ts = str(datadescription.GetTimeStep())
    SaveData('grid_compressed_' + ts + '.tucker', proxy=vtk_uniq_test_name_9vtm)
```

- ParaView Plugin can be loaded into the ParaView GUI or Catalyst in-situ and contains a reader and writer
- Plugin maps between ParaView and TuckerMPI data structures in parallel, handles MPI communicator re-mapping, and supports input types of vtkMultiBlockDataSet, vtkImageData, and vtkStructuredGrid
- Writer compresses vtkMultiBlockDataSet input by iterating over all blocks in parallel and creating a TuckerMPI compressed file on disk for the block
- Reader reconstructs in parallel a vtkMultiBlockDataSet from compressed directory blocks
- Minimal changes required to add compression writer to existing Catalyst coprocessor pipelines.

Wavelet Data Set

- Synthetic wavelet data set, 3 million cells, 15 variables.
- Representative of SPARC (SNL ATDM app) benchmark data set.
- Parallel compression on 16 MPI ranks (ParaView in client/server mode).
- Compression of ~2000X for error threshold of 1%.



Comparison of the compressed (left) vs the original fields (right) of a wavelet data set (3 million cells, 15 cell variables).

Ongoing/Future Work

- In-situ reader (i.e. reconstruction) plugin.
- User-specified partial reconstruction.
- In-situ analysis on compressed data.
- Compress time snapshots in streaming manner.
- Compression of unstructured mesh data.

References

[1] W. Austin, G. Ballard & T. G. Kolda, "*Parallel Tensor Compression for Large-Scale Scientific Data*," IEEE IPDPS, Chicago, IL, 2016, pp. 912-922.

[2] N. Fabian, *et al.*, *"The ParaView Coprocessing Library: A scalable, general purpose in situ visualization library"*, IEEE LDAV, 2011, pp. 89-96.

U.S. DEPARTMENT OF ENERGY
NNSA National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SAND No. _____

ECP EXASCALE COMPUTING PROJECT

Sandia National Laboratories