Analyzing the Performance and Accuracy of Lossy Checkpointing on Sub-iteration of NWChem

Tasmia Reza[1,2], Kristopher Keipert[2], Sheng Di[2], Xin Liang[2,3], Jon C. Calhoun[1], Franck Cappello[2]
[1]Clemson University, Clemson, SC, USA
[2]Argonne National Laboratory, Lemont, IL, USA
[3]University of California, Riverside, CA, USA

SC19 Denver, CO | hpc is now.

#ClemsonSC19

CLEMSON UNIVERSITY

Introduction

- Future exascale systems are expected to experience more frequent failures.
- As a result, reliance on checkpoint-restart increases.
- This increases the computational cost for running applications.
- Significant progress is lost between checkpoints.
- Lossless and lossy compression are used to reduce volume of data storage and increase efficiency of memory bandwidth.
- Lossy compression reduces data volume by order of magnitude compared to lossless compression.
- However, lossy compression adds user controlled inaccuracies into the data.

Motivation

- HPC applications with large-scale iterative methods are good examples to test the lossy compression because
  - It converges to correct result at the end of computation even after injecting inaccurate lossy data.
  - It takes large volume of computation and significantly longer time to run.
  - Iterative methods are common to many HPC applications.

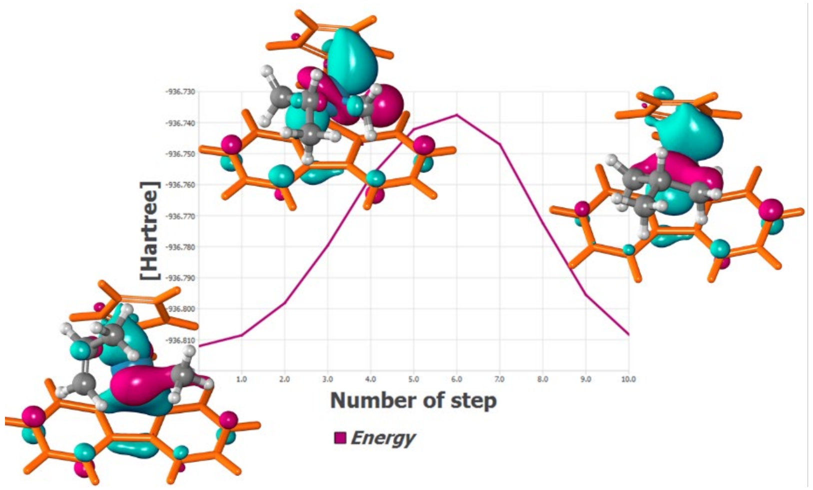
- This work was done under funding of Exascale Computing Project (ECP) to test the efficiency of lossy compression in a quantum chemistry application.

ECP EXASCALE COMPUTING PROJECT

NWChem

- NWChem is an open source computational chemistry / materials package
  - Optimized for high-performance parallel supercomputers
  - Rich suite of methods
    - **DFT** (Gaussian/Plane Wave)
    - **Many-body methods** (CI, CC, MBPT :: TCE)
    - **Molecular dynamics** (*ab initio*, AMBER/CHARMM Force Fields)
  - Actively developed since 1994
    - PNNL Environmental Molecular Sciences Laboratory
- It offers a broad array of molecular modeling capabilities that can be deployed on all the major supercomputing platforms.
- Push the limits of compute and I/O throughput

EMSL



https://github.com/nwchemgit/nwchem

Coupled Cluster Singles and Doubles (CCSD)

Coupled cluster - "Gold Standard" methods
Solve cluster amplitudes that are used to approximate the exact wave function.

An exponential ansatz of the wave function of this cluster is

$$|\Psi\rangle = e^{T} |\Phi_0\rangle$$

The cluster operator is,
$T = T_1 + T_2 + \ldots\ldots\ldots + T_N$

where, T_1 is the operator of all single excitations, T_2 is the operator of all double excitations and so forth.

- CCSD NWChem takes around 20 iterations to solve.
- It requires large memory/disk space due to expensive computation scaling $O(N^6)$.
- Improved NWChem performance ⇨ tackle larger problems
- Lossless compressions ≈ 1.2x

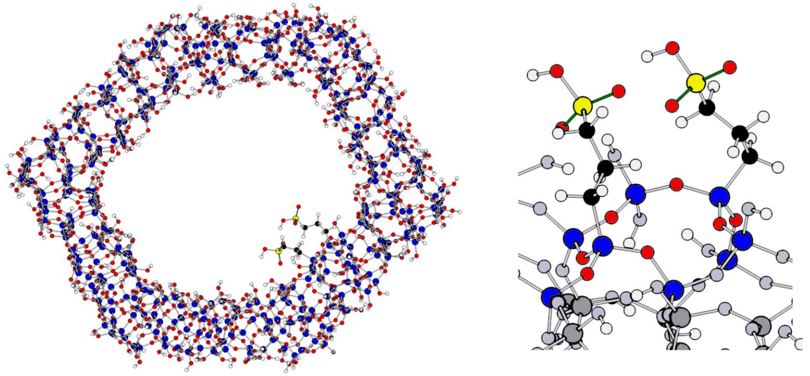
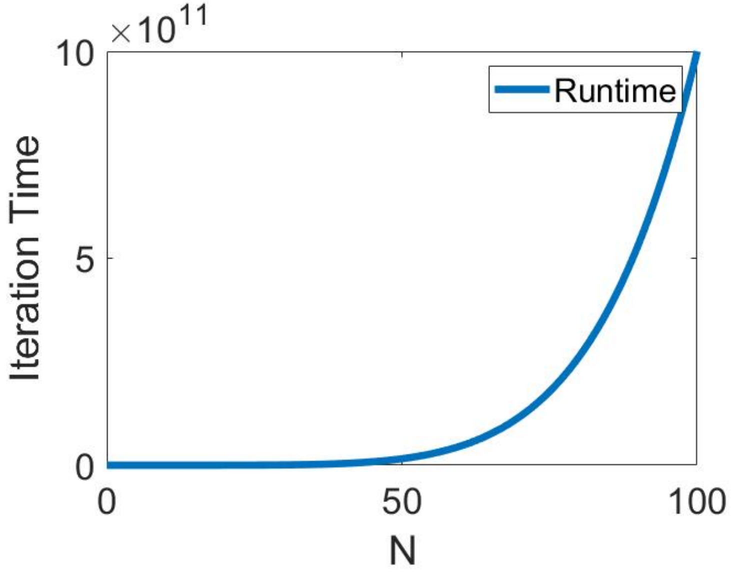


Image: Acid-functionalized mesoporous silica catalyst

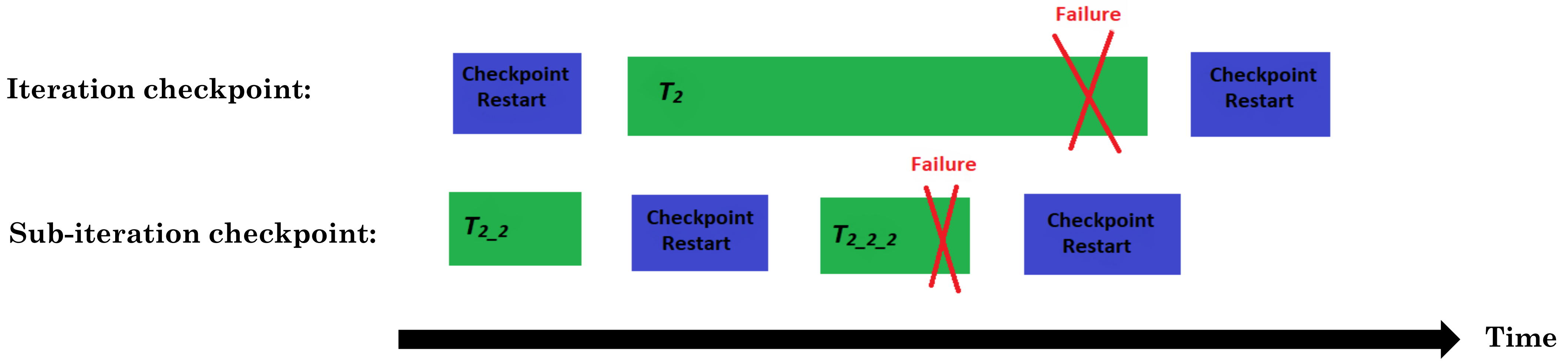
Sub-Iteration Checkpointing of NWChem

- Previous NWChem checkpointing work focus on the coupled-cluster singles and doubles (CCSD) computation at a per-iteration granularity.
- The per-iteration time can be significantly high even consuming hours or even days.
- To address this large overhead when restarting, we target checkpointing at a sub-iteration level.



Sub-Iteration Checkpointing of NWChem

- The T_2 tensor can be broken into 24 intermediate sub-tensors.
- The NWChem is modified to checkpoint each sub-tensor individually/incrementally.
- It is possible to recover at a sub-tensor granularity.
- In this case, we are only ever losing work on the current sub-tensor.



SZ – Error Bounded Lossy Compressor

- SZ provides multiple error bounding modes with user controllable error bound.

- Compression ratio 10x - 100x.

- Supporting I/O library or data formats:
  - HDF5
  - ADIOS
  - NetCDF

- Collaborations:
  - ExaSky (HACC, NYX)
  - GAMESS
  - NWChemEx
  - CODAR etc.

Argonne NATIONAL LABORATORY

SZ integration into NWChem application

- Code structure of calling SZ-C interface from NWChem

```
 //X is sub-tensor data
if iteration number == 10 //Simulate lossy restart
                X’= compress(X)
                X = decompress(X’)
end if
```

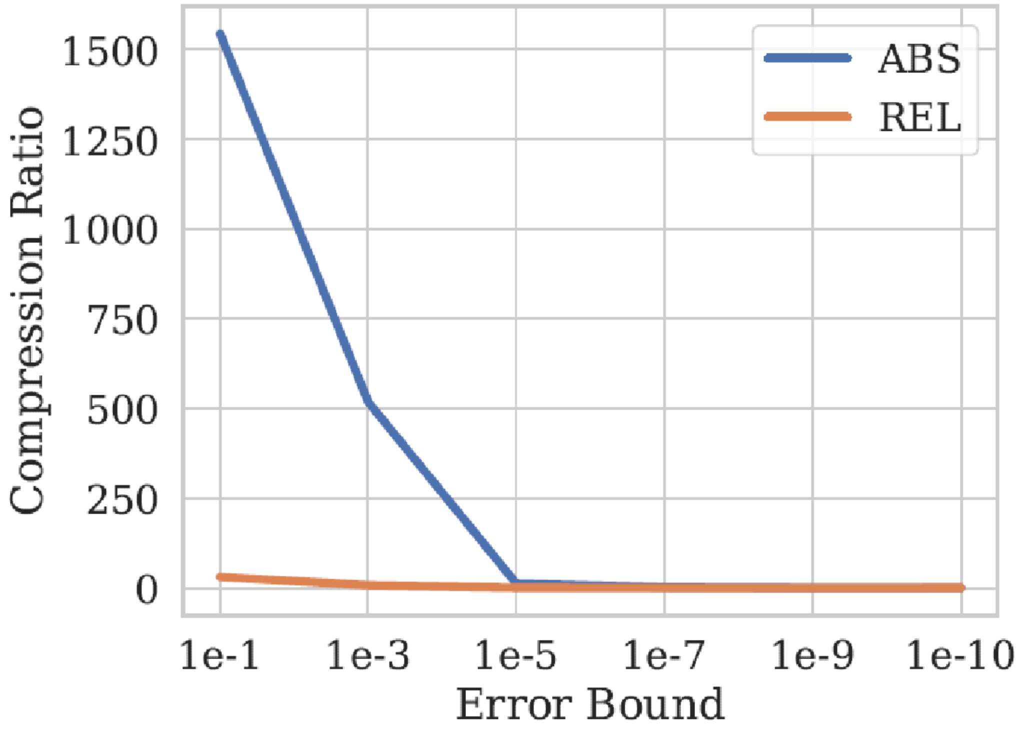
- The input data, data dimension and numeric error bound value are the inputs in the SZ-C interface.
- The decompressed data is collected as the output.
- Error bounds tested:
  - Relative
  - Absolute
  - Values: 1E-1,1E-3, 1E-5, 1E-7, 1E-9 and 1E-10.
- The validation of acceptable energy deviation value as 1E-5 was done by quantum chemistry expert.

Experimental Setup

- Experiments were run on the Bebop Cluster operated by the Laboratory Computing Resource Center (LCRC) at Argonne National Laboratory.
- Bebop nodes consist of Intel Xeon E5-2695V4 CPUs with 128GB DDR4 RAM.
- The lossy checkpointing experiment at the sub-iterations of NWChem uses simulation of water molecules.
- The simulation converges in 17 iterations.
- NWChem Version: 6.8.1 release.
- SZ version: 2.1.5.

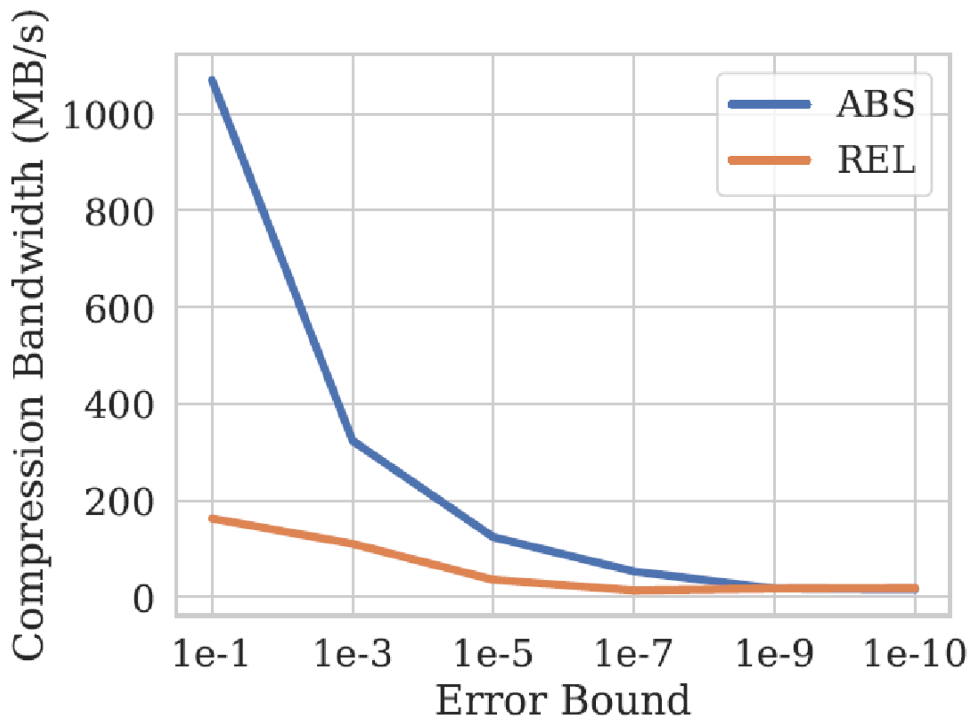
Average Compression Ratio of Individual T_2 Sub-Tensors

- The absolute error bounds show significant higher compression ratios than relative error bounds for equivalent error bounds.
- Relative error bounds show 1-2x compression ratio for all the error bounds.
- Absolute error bound performs better until 1E-7.



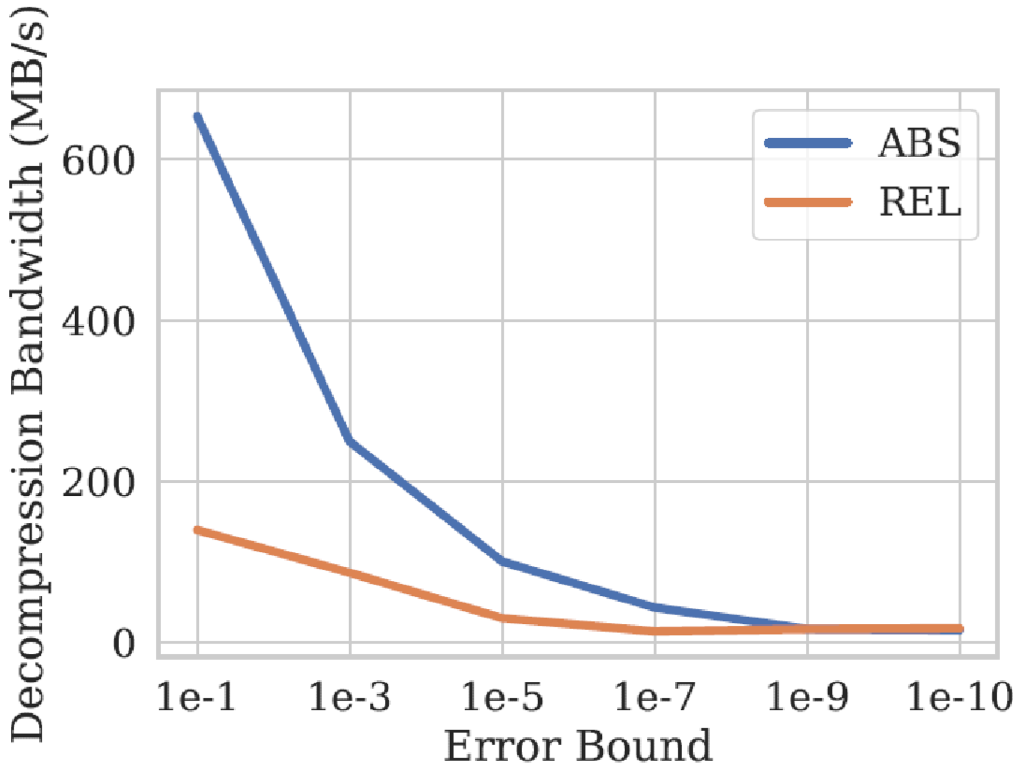
Average Compression Bandwidth of Individual T_2 Sub-Tensors

- The absolute error bound yield higher compression bandwidth compared to the equivalent configuration of relative error bound.
- Increasing error bound shows increased compression bandwidth for both error bounding types.
- The lower error bounds enforce more accurate data and the compression bandwidth approaches zero indicating that compression yields unacceptable performance.



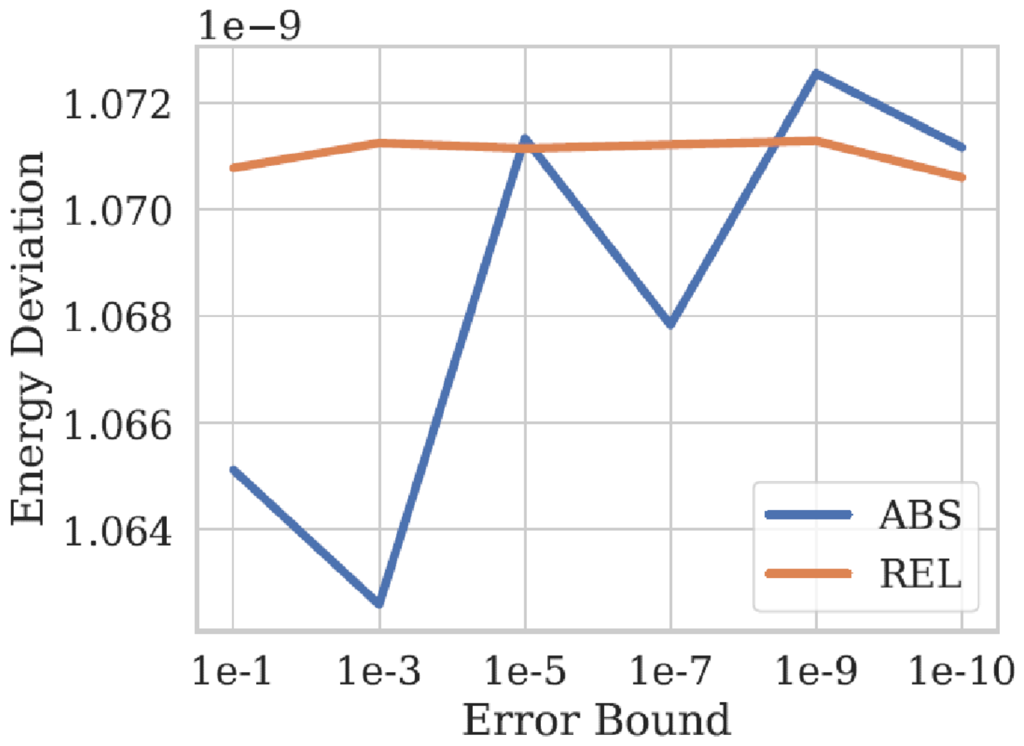
Average Decompression Bandwidth of Individual T_2 Sub-Tensors

- Decompression bandwidth show similar behavior to compression bandwidth.
- The major difference is that the decompression bandwidth is lower than the compression bandwidth for similar high error bounds.
- As the error bound better preserves the data the decompression bandwidth approaches zero which can lead to increased overhead for NWChem simulations with large quantities of data.



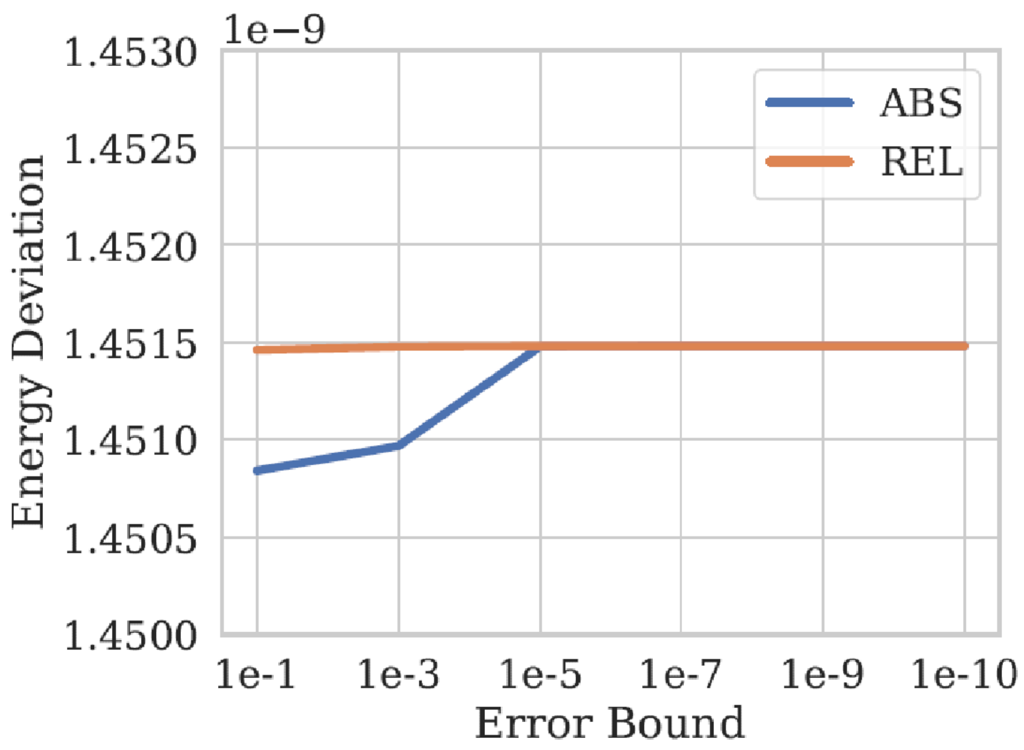
Average Energy Deviation of Compressing Individual T_2 Sub-Tensor

- The deviation in energy is very minor which is 1E-9 for all configurations and it is well below the level of acceptability of 1E-5.
- It is possible to do lossy-compressed checkpointing each sub-tensor of T_2 and successfully restart.
- All runs finish with at most 1 extra iteration.



Energy Deviation for Compressing All Sub-Tensors Simultaneously

- Compared to previous plot it is noted that the deviation is slightly higher indicating that there is more deviations in the simulation.
- This is due to be restarting from a lossy checkpoint where all the sub-tensors are lossy compressed.
- Even with the higher magnitude deviation, it is well within the simulation accuracy bound of 1E-5.
- This shows that sub-iteration checkpointing is feasible to enable restarting when failure strikes and does not impact the accuracy of an NWChem simulation.



Conclusion

- We propose sub-iteration level checkpointing at NWChem.
- The results show that absolute error yields better performance than relative error for error bounds on the range 1e-1 to 1e-5.
- For all the experiments, the number of extra iterations increased by at most 1 compared to lossless run.
- The energy deviations were lower than our accuracy cut-off indicating the sub-tensor level lossy checkpointing is acceptable in NWChem application.

Future Work

- Future work will study the impact of restarting multiple iterations from lossy compressed checkpoints, evaluate the resultant energy deviation and number of iterations.

- In addition, it is planned to derive and verify a performance model for lossy checkpointing NWChem to understand the trade-off between the number of extra iterations upon restart and time saved by lossy compressed checkpointing.

CLEMSON UNIVERSITY

Acknowledgement

- This work is supported by the National Science Foundation under Grant No. SHF-1910197 and Grant No. SHF-1619253.

- It is supported by the US Department of Energy under subaward No. 9F-60179.

Acknowledgement

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.

EXASCALE COMPUTING PROJECT

Questions?

Thank you