ECCO PROJECT MEETING 2018

Argonne NATIONAL LABORATORY

COMPRESSING CHECKPOINTS IN MITGCM ADJOINT COMPUTATIONS

KAIYUAN HOU
Department of Electrical Engineering & Computer Science
Northwestern University
Evanston, IL, USA

PAUL HOVLAND
SRI HARI KRISHNA NARAYANAN
BOGDAN NICOLAE
Mathematics & Computer Science Division
Argonne National Laboratory
Argonne, IL, USA

NAVJOT KUKREJA
Department of Earth Science & Engineering
Imperial College London
Kensington, London SW7 2AZ, UK

DANIEL GOLDBERG
The University of Edinburgh
Edinburgh, United Kingdom

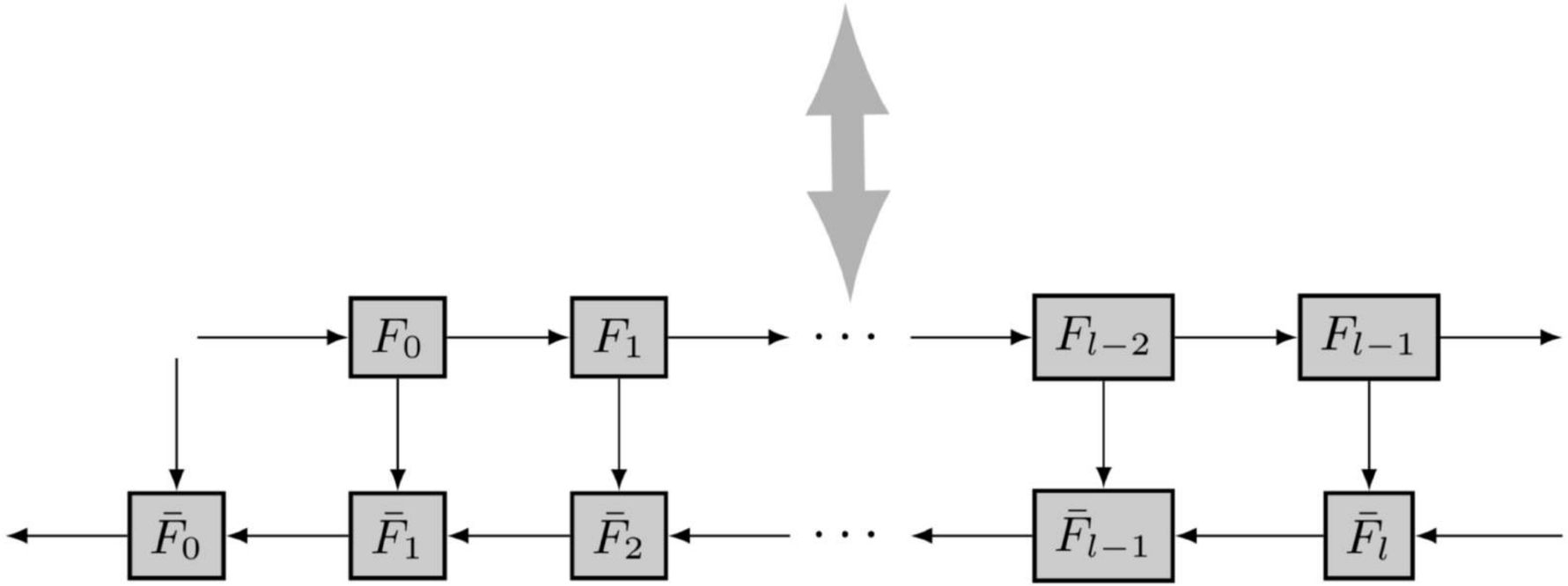
OUTLINE

- Review of checkpointing in adjoint computations
- Optimal checkpointing strategies
- Checkpointing in OpenAD/MITgcm
- Checkpoint compression
- Next steps

CHECKPOINTING IN ADJOINT COMPUTATIONS

Adjoint time steps require intermediate states in reverse order from computation during forward sweep (recompute or store)

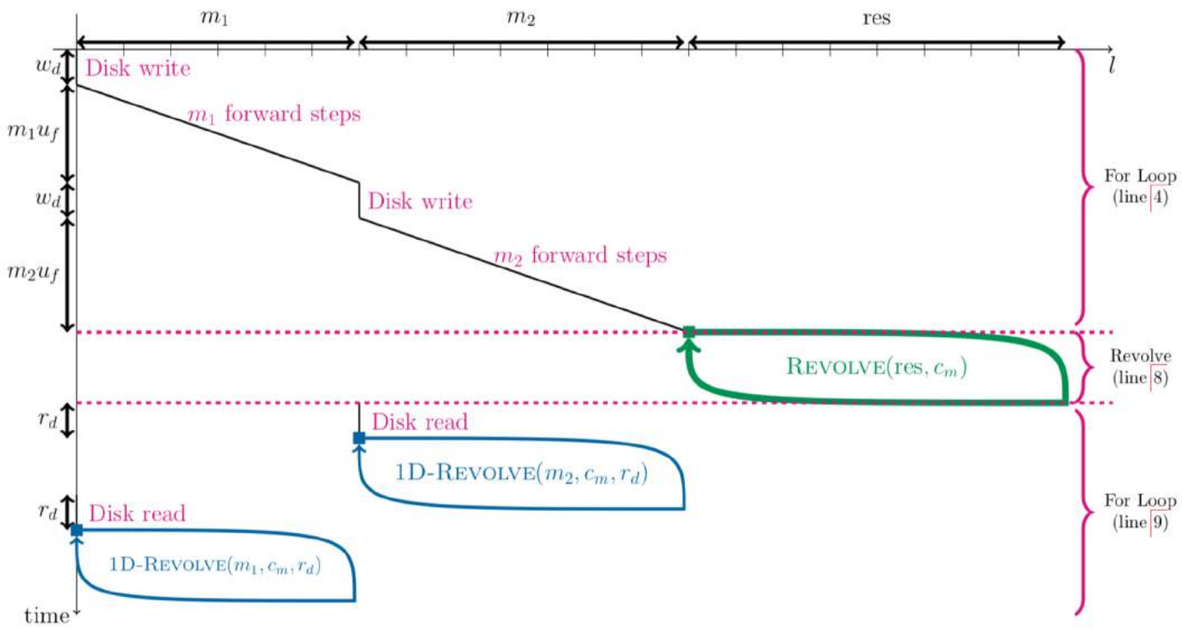
$$F_i(x_i) = x_{i+1} \quad i < l$$
$$\bar{F}_i(x_i, \bar{x}_{i+1}) = \bar{x}_i \quad i \leq l$$



OPTIMAL CHECKPOINTING STRATEGIES

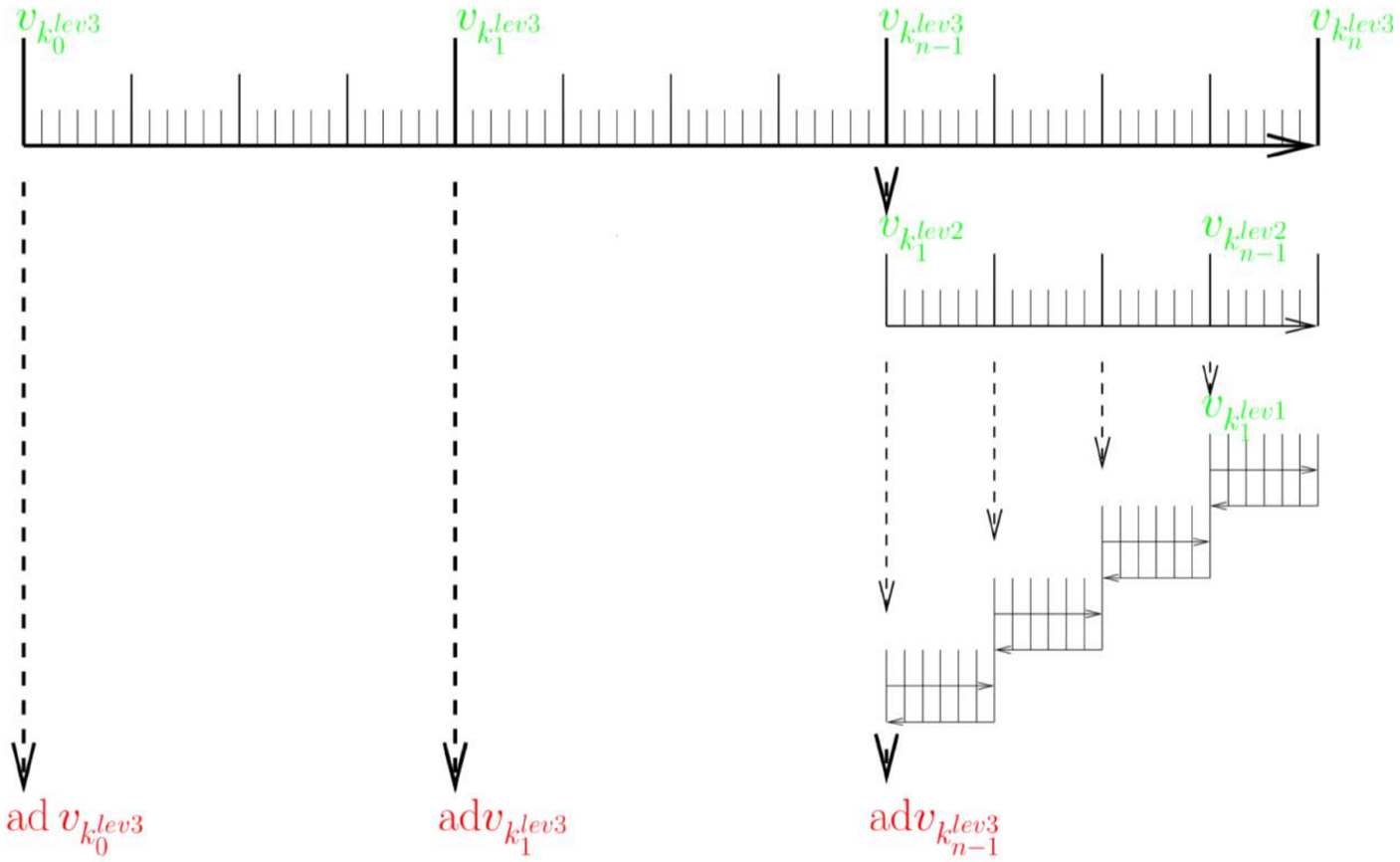
Trade off recomputation against storage

- Binomial checkpointing: optimal if number of checkpoints is limited, time to write/ read a checkpoint is negligible
  - Griewank &Walther, *Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation*
- Periodic multilevel checkpointing: optimal if numberof checkpoints is unlimited, time to write/read checkpoints is nonnegligible
  - Aupy & Herrmann, *Periodicity in optimal hierarchical checkpointing schemes for adjoint computations*
  - Schanen et al., *Asynchronous Two-level Checkpointing Scheme for Large-scale Adjoints in the Spectral-element Solver Nek5000*



OTHER CHECKPOINTING STRATEGIES

- Equidistant:
  checkpoint every N timesteps
- Hierarchical:
  checkpoint every N_i timesteps within level i
- Binary:
  checkpoint at midpoint of each level
- None of these are “optimal” but easy/easier to implement



CHECKPOINTING IN OPENAD/MITGCM

- Binomial checkpointing to disk
- Use all available memory to reduce cost of an adjoint step (tradeoffs in subroutine-level checkpointing versus recomputation)
- Probably not optimal
- Works well in practice

CHECKPOINT COMPRESSION

- Increase the number of checkpoints that can be stored through compression
- Reduce time to write/read checkpoints through compression
- Lossless compression
  - In principle, identical computation to no compression
  - Tradeoff between time to compress/decompress and savings in time to write/read, number of available checkpoints
- Lossy compression
  - Sacrifice accuracy of checkpoints for better compression ratios
  - Tradeoff among time to compress/decompress, savings due to compression, and accuracy of gradient computation
- **WORK IN PROGRESS: PRELIMINARY RESULTS**

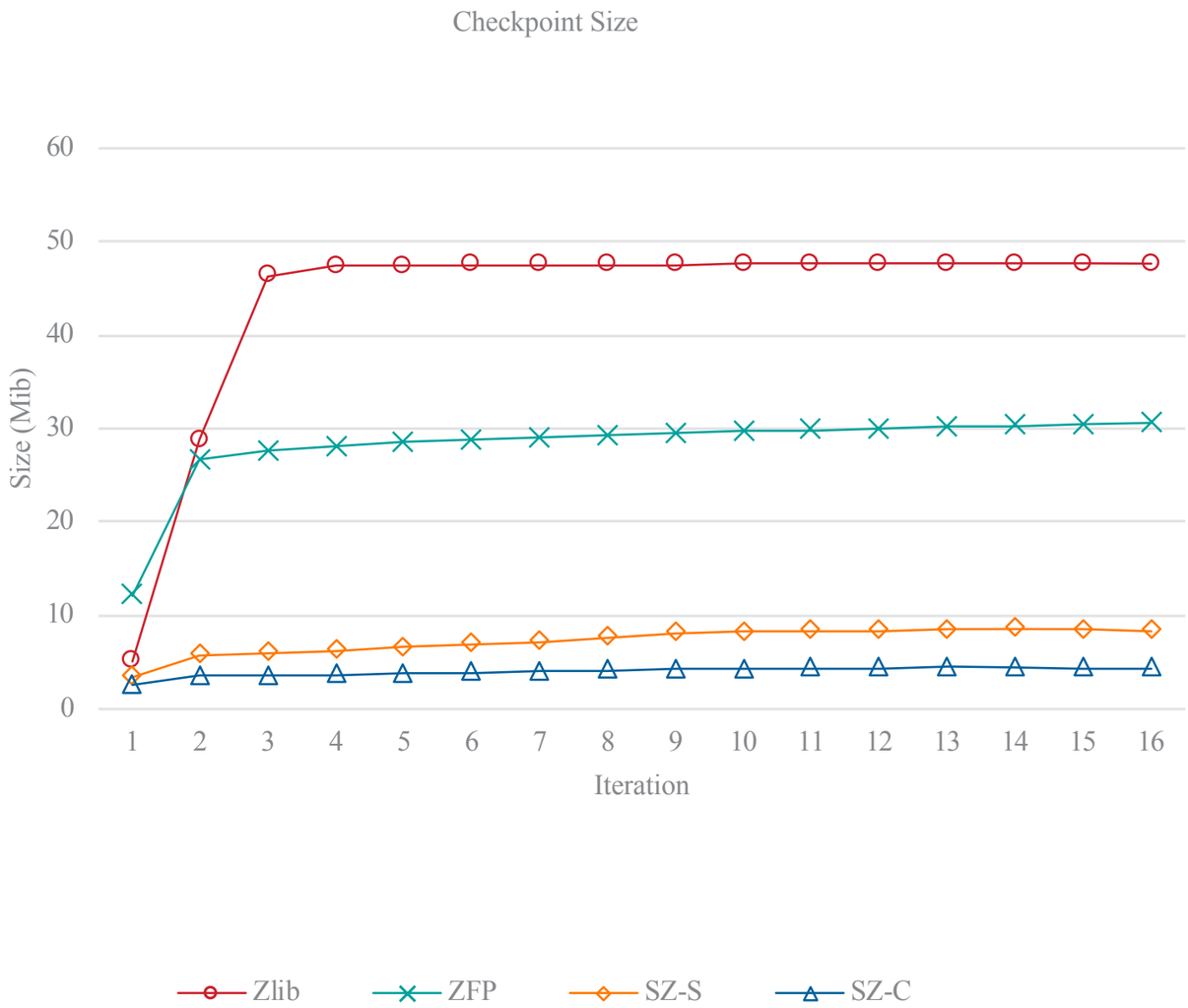
ACHIEVED COMPRESSION FOR HS94.1X64X5

Original checkpoint: 436MiB

Zlib: lossless

Zfp: lossy (10^{-4} tol)

SZ: lossy (10^{-4} tol)



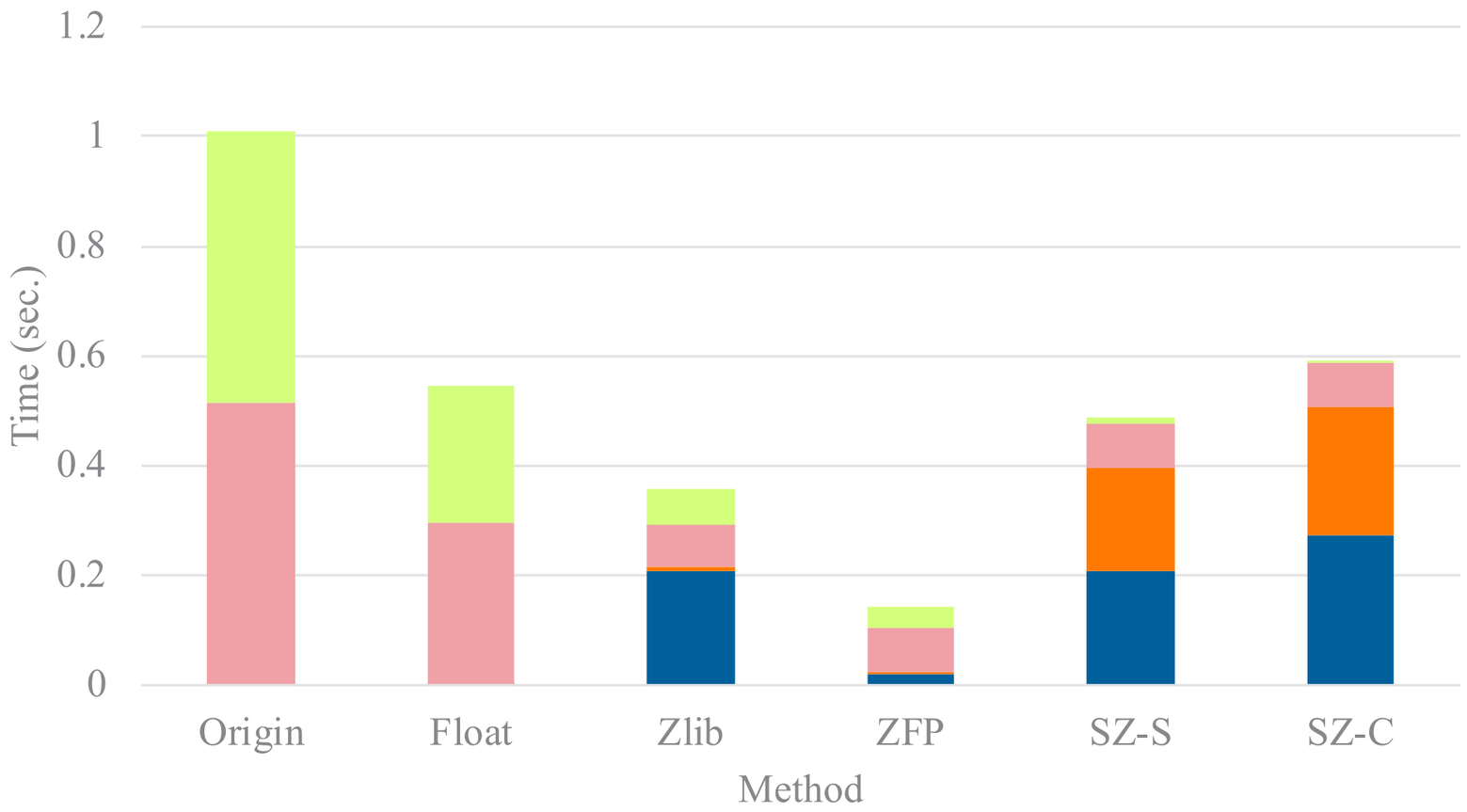
TIME TO WRITE AND READ A COMPRESSED CHECKPOINT

Float: lossy (32bit)

Zlib: lossless

Zfp: lossy (10^{-4} tol)

SZ: lossy (10^{-4} tol)



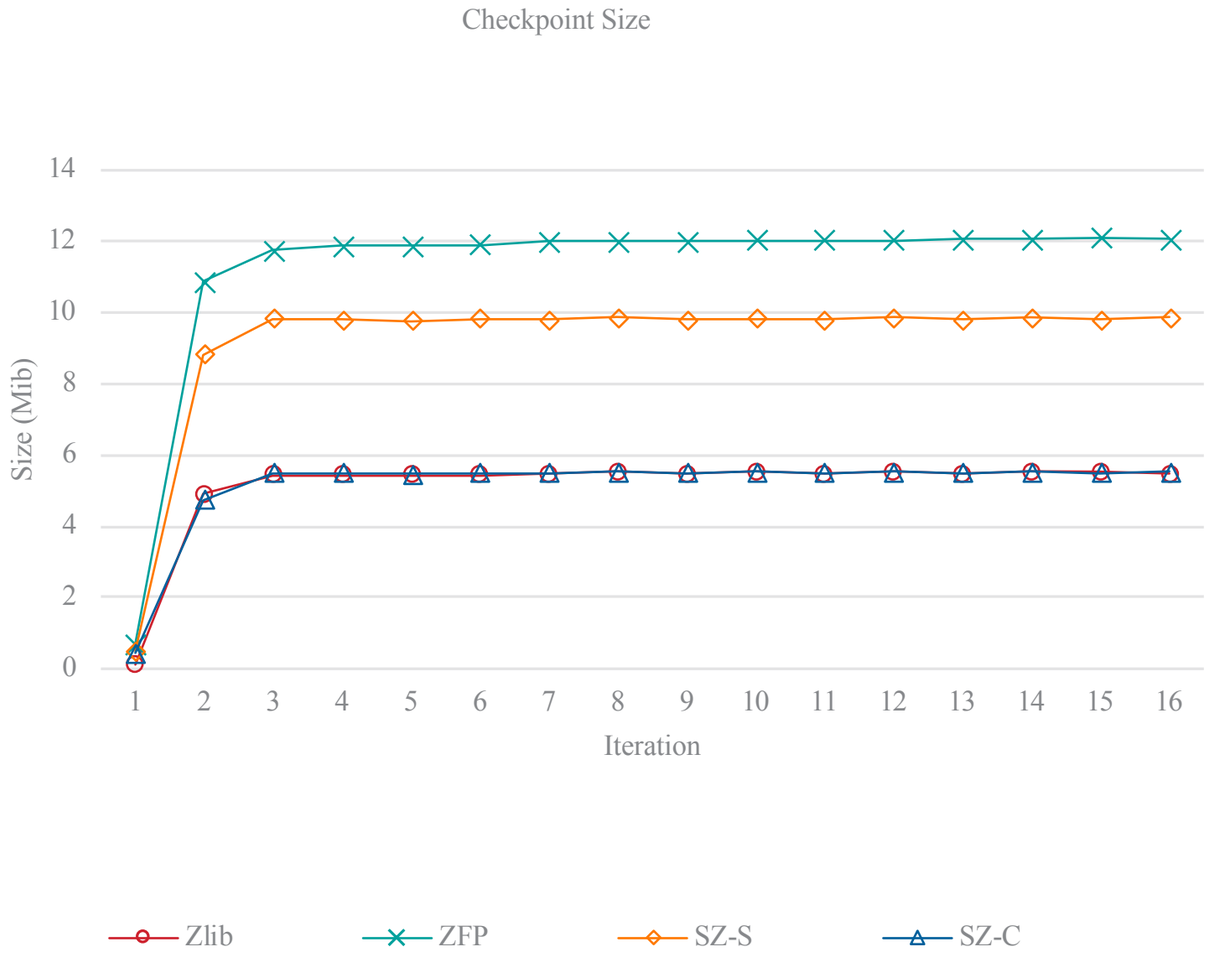
ACHIEVED COMPRESSION FOR HALFPIPE_STREAMICE

Original checkpoint: 31 MiB

Zlib: lossless

Zfp: lossy (10^{-4} tol)

SZ: lossy (10^{-4} tol)



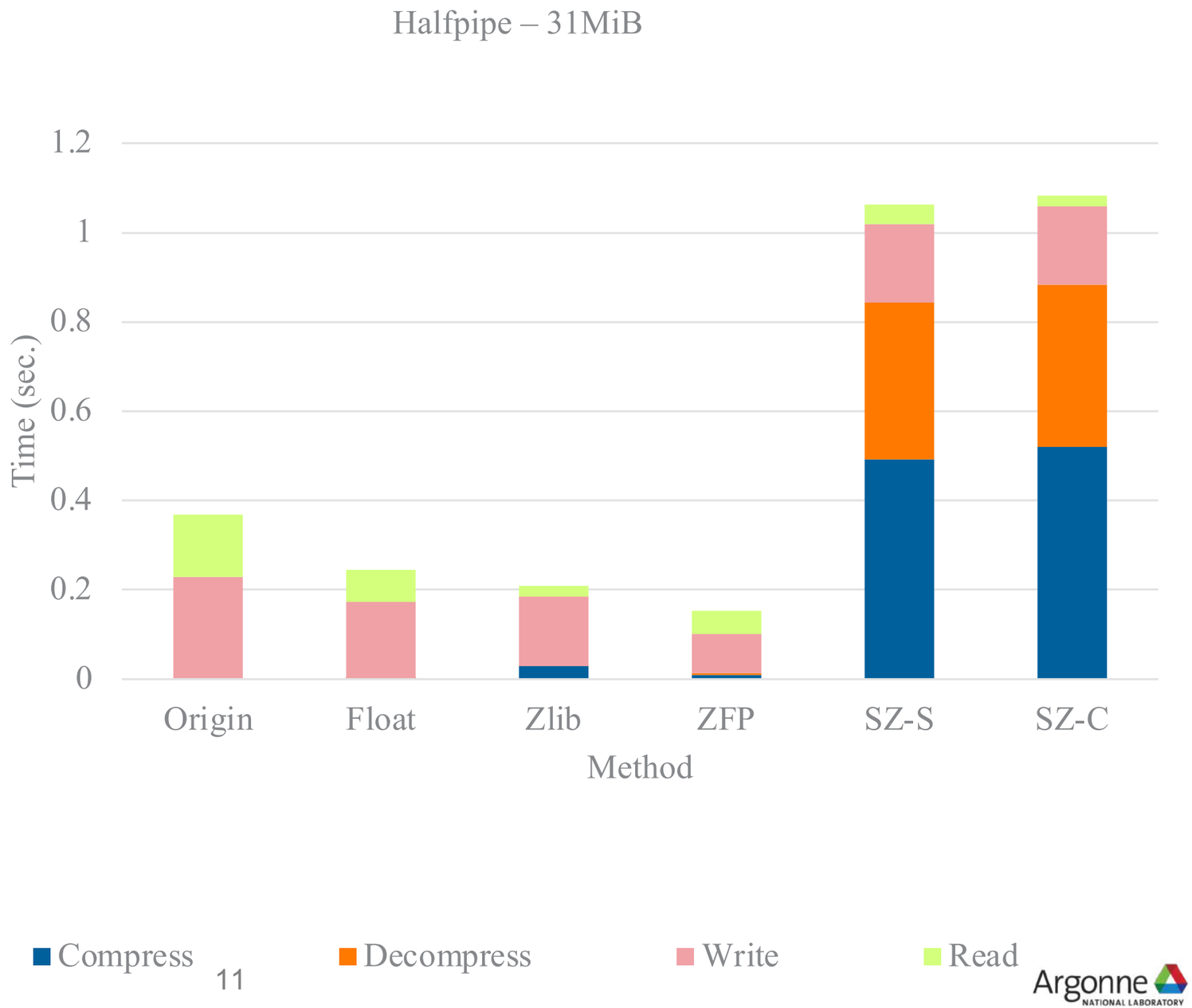
TIME TO WRITE AND READ A COMPRESSED CHECKPOINT

Float: lossy (32bit)

Zlib: lossless

Zfp: lossy (10^{-4} tol)

SZ: lossy (10^{-4} tol)



END-TO-END EXECUTION TIMES

- Significant performance improvement for hs94 using Zlib (lossless)
- No improvement for halfpipe (I/O time negligible compared to timesteps)

Program	Origin	Float	Zlib	ZFP	SZ-S	SZ-C
hs94	104.2	78.8	59.4	54.7	71.9	72.0
halfpipe	807.2	831.4	804.7	832.5	897.2	894.0

NEXT STEPS

- Work with SZ developers to understand performance results
- Analysis of errors induced in gradient (preliminary results suggest errors in gradient commensurate with errors in checkpoints)
- Examine other MITgcm configurations, other adjoint computations
- Revisit assumptions in OpenAD/MITgcm checkpointing
  - Would periodic multilevel be better than binomial?
  - Should some of the checkpoints be in memory?
  - What is the effective limit on the number of disk checkpoints?
- Compression of in-memory checkpoints
- Optimal checkpointing strategy for fixed/variable compression ratios