# Lossy Compression for Checkpointing: Fallible or Feasible?

*Xiang Ni and Laxmikant Kale, Department of Computer Science, University of Illinois at Urbana-Champaign*
*Tanzima Islam, Kathryn Mohror and Adam Moody, Lawrence Livermore National Laboratory*

As HPC applications scale to hundreds of thousands of processors and higher, large checkpoints not only consume a lot of space, making it costly to fit them in memory or burst buffers, but it also takes a significant amount of time to transfer them to stable storage. To address this challenge, we propose using lossy compression to reduce checkpoint size and studying the trade-off between the loss of precision and the compression ratio.

## Goals

- Given the failure rate of a machine, how lossy can the checkpoints be (determining the compression ratio)?
- If the application suffers from multiple failures, does it matter how they are spread throughout the run?

## Approach

### Compression technique to reduce checkpoint size

- For scientific applications, lossless compression typically reduces the checkpoint size by 15-50% .
- With lossy compression, the compression ratio can be improved to 3-5x.

### Selective lossy compression

```
checkpoint function()
compression_on(myNumParticles, 32)
store, x_position
store, y_position
store, z_position
compression_on(myNumParticles, 16)
store, x_velocity
store, y_velocity
store, z_velocity
compression_off()
store, iType
```

Lossy level control.

Selective precision level for different types of data. We found this is useful in our experiments: for certain runs of ChaNGa, compressing position fields with lossy level 16 causes program to hang.

Turn compression off: do not compress control data.

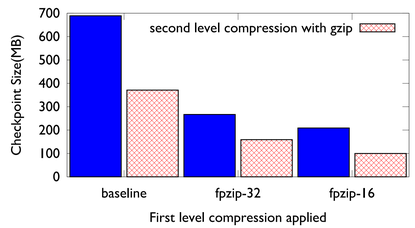
**Fig. 1**

## Preliminary Results

### Experimental setup

- Application: ChaNGa, an N-body cosmology simulation
- Dataset: dwf1
- Lossless compression: gzip
- Lossy compression: fpzip, a lossy compression library based on predictive coding and quantization.

### Compression ratio

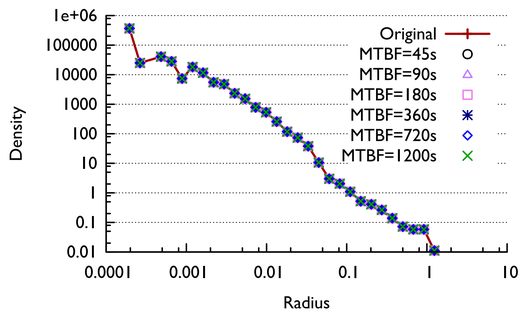



| | Compression Ratio |
|---|---|
| gzip | 1.85 |
| fpzip-32 | 2.58 |
| fpzip-32 + gzip | 4.33 |
| fpzip-16 | 3.29 |
| fpzip-16 + gzip | 6.89 |

**Fig. 2**

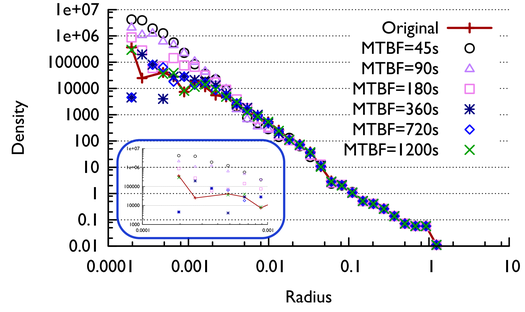
### Effect of failure rate

We examined the dark matter density profile at the end of the run to verify correctness. We injected failures using a uniform distribution through one-hour run of ChaNGa.




Using fpzip 32-bit compressor:
32 least significant bits are discarded.

**Fig. 3**



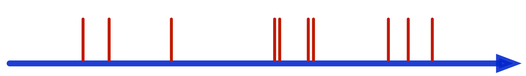

Using fpzip 16-bit compressor:
48 least significant bits are discarded.

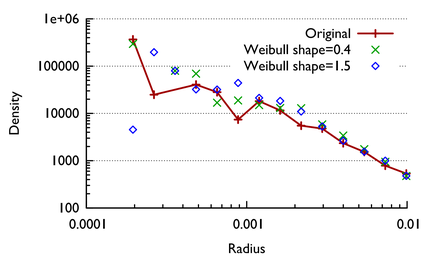
**Fig. 4**

### Effect of failure distribution

Weibull distribution
shape parameter 0.4, decreasing failure rate.

shape parameter 1.5, increasing failure rate.






fpzip 16-bit compressor is used. ChaNGa is more sensitive to failures at the later stage of the simulation.

**Fig. 5**

## Future Plan

- Extend to different application classes, like linear algebra, chaotic applications.
- Study the mixture of lossy and lossless compression based on semantic information.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-POST-658374.