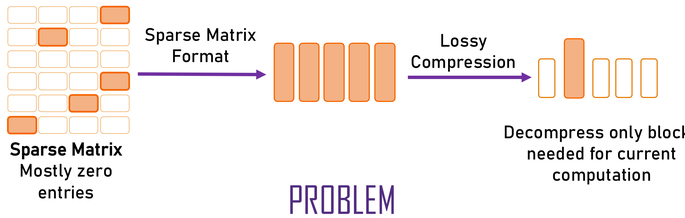
# Performance Analysis of Inline Compression on Sparse Matrix Vector Multiplication

Author: Precious Eyabi
Email: peyabi@clemson.edu
Advisor: Dr. Jon Calhoun

## MOTIVATION

» **Sparse matrix vector multiplication** (SPMV) plays a **critical role in scientific computing,** from computational fluid dynamics, to artificial intelligence, to economics problems, the use cases are vast
» **The data required for** these high performance computing (HPC) jobs is **quickly outgrowing available memory storage**
» Inline compression [1,2] aims to mitigate this memory bottleneck by **keeping data compressed in storage and only decompressing as needed**
» Traditional SpMV-specific compression methods excel with highly sparse matrices, but for sparse matrices with **a high number of non-zero entries, further compression is needed** to see significant improvements in memory footprint

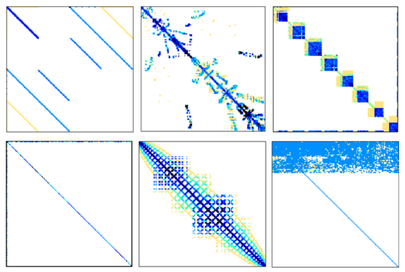



## PROBLEM

How do we balance the memory savings of inline compression against its performance cost in sparse matrix-vector multiplication?

## MATRICES

» Matrices were chosen to **represent a diverse set of real-world problems** that require sparse matrix-vector multiplication (SpMV). Their differing structures and numerical properties enable analysis of how matrix characteristics influence performance.



» **Problems represented include**:
- Case39 - Power Network Optimization
- Bcsstk13 - Computational Fluid Dynamics
- Nxp1 - Circuit Simulation
- Bundle_adj - Computer Vision
- Ga3As3H12 - Theoretical/Quantum Chemistry
- Poli4 - Economics Problem

Pictured above are images of the matrices used for our tests. Matrices were obtained from SuiteSparse Matrix Collection [3]

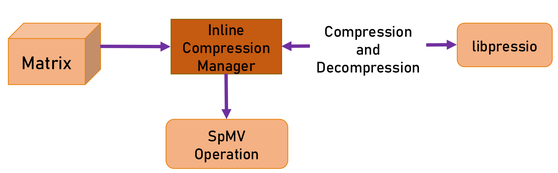
## EXPERIMENTAL SETUP

» We analyze the effects of SpMV on a sparse matrix in three forms, coordinate format (**COO**), compresses sparse row format (**CSR**), and compressed sparse column format (**CSC**)
» We apply optimizations to examine their **effects on execution time and the overall memory footprint** with the goal of understanding where the biggest performance bottlenecks are and how to mitigate them

» **Metrics**
- Compression Ratio
- Time (us)

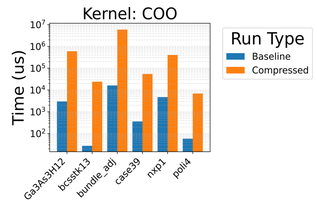
» **Hardware**
- Experiments were performed on **Clemson's Palmetto 2**
- CPU: **AMD Genoa (EPYC 9654)** with 96 cores

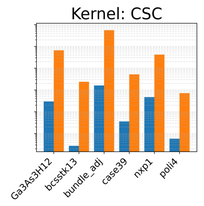
» The **lossy compressor SZ** [4] was used for compressing the non-zero entries of the sparse matrices
» **Error bound: 0.001**
» Pictured is a diagram of the workflow, the inline compression manager uses libpressio to implement SZ

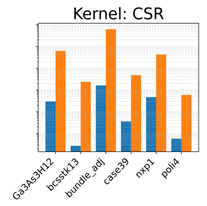



## RESULTS

We examine how the performance of the SpMV kernels is affected by introducing inline compression. The results show that matrix composition has a greater impact on the memory savings achieved through compression than matrix size alone. Then, we examine how compression performance is influenced by the compressor's configuration, specifically by comparing configurations with and without caching.




» The performance drop from baseline to compressed SpMV isn't a compute problem but rather a memory and latency problem
» This is the same reason that HPCG achieves a tiny fraction of a system's HPL number even though it's running on the exact same hardware

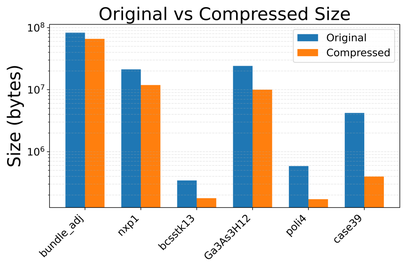
**LinShine[5]**

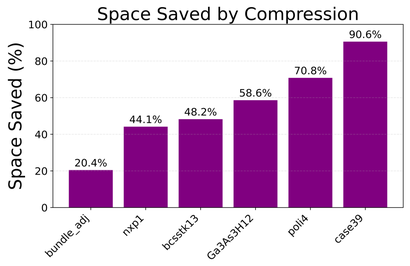
**1.0%**
Retention of its HPL number

**SpMV w/ Compression**

**~0.84-1.18%**
Retention of baseline performance

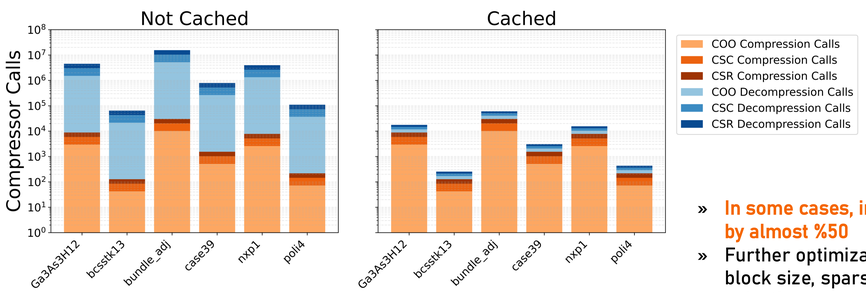
» **This comparison shows that the observed performance degradation is consistent with the behavior of memory-bound benchmarks such as HPCG, where throughput is similarly constrained by memory access rather than computational capability**



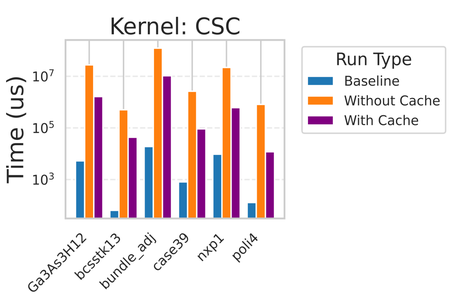

» The matrix that achieved the highest compression ratio was **case39** demonstrating the significant impact that compression can have
» The matrix that benefited the least was **bundle_adj**, the largest matrix
» The improvements achieved through compression are not directly tied to the size of the dataset, but rather to the characteristics and composition of the data itself

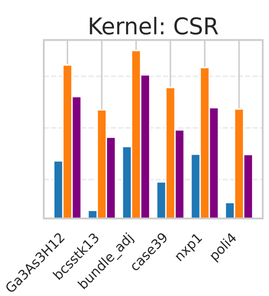
**Plots are in log scale for visibility**

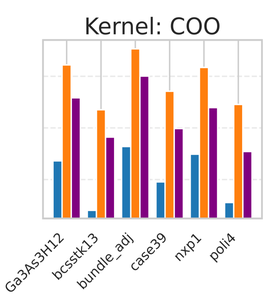



» **This section examines the effects of our proposed optimizations**
» By utilization caching, we were able to decrease the number of decompression calls which in turn reduced overhead and improved run time

» **In some cases, including caching reduced run time by almost %50**
» Further optimizations, such as tuning compressed block size, sparsity pattern aware blocking, and selecting appropriate compressors, could help mitigate this performance loss and will be explored in future work




## CONCLUSIONS

With system storage being **limited** and **expensive**, minimizing the memory footprint of your application is imperative. In this analysis we have found that:

» It is possible to achieve **significant reductions in the memory footprint** through inline compression, but it will impact the overall performance
» While the performance collapse is within an **expected range**, careful considerations need to be made to ensure that significant levels of performance are not lost
» The impact of inline compression is **workload dependent** so the characteristics and composition of the sparse matrix matters

## FUTURE WORK

» This is in in-progress work, we plan to further examine **other factors that affect performance**, such as **sparsity pattern aligned** block sizes, or **type of lossy compressor**, to determine which conditions best minimize the performance loss
» We also plan to **extend the current analysis** to evaluate **lossless compression of the index structures**, rather than focusing solely on compressing the matrix data, with the goal of seeing how much memory saving can be achieved
» Finally, we plan to investigate the **energy efficiency** of inline compression in HPC systems. Since **data movement** is a significant contributor to energy consumption, **reducing the volume of transferred data** through compression may improve energy efficiency. This analysis will **quantify the impact of inline compression on energy usage** and evaluate the associated performance and energy tradeoffs

## REFERENCES

[1] Emily Lattanzio, Sansriti Ranjan, Robert Underwood, Thomas Saupe, Robert Speck, and Jon C Calhoun. 2025. Performance Analysis of Inline Compression in pySDC. In 2025 IEEE High Performance Extreme Computing Conference (HPEC). IEEE, 1–8.
[2] Steven Platt and Jon C Calhoun. 2024. Analyzing an In-line Compression on the Matrix Matrix Multiplication Kernel. In 2024 IEEE High Performance
Extreme Computing Conference (HPEC). IEEE, 1–11.
[3] https://sparse.tamu.edu/
[4] Di, Sheng, and Franck Cappello. "Fast error-bounded lossy HPC data compression with SZ." 2016 ieee international parallel and distributed processing symposium (ipdps). IEEE, 2016.
[5] https://top500.org/lists/hpcg/2026/06/

Get in Contact