Modeling Power Usage for the SZ Lossy Compressor on HPC Systems

GRANT WILKINS and JON C. CALHOUN (ADVISOR)*, Holcombe Department of Electrical and Computer Engineering, Clemson University

Lossy compressors are becoming more prevalent due to the increasing volumes of data produced in HPC systems. Compressors are integral to computational workflows to optimize data I/O: transporting smaller amounts of data is more time and energy efficient. By modeling power consumption of lossy compressors, we work towards improving energy efficiency. In this paper, we present a characterization of the power usage of the SZ lossy compressor over several datasets. We describe the piecewise energy consumption of SZ's internal processes, while also computing the total energy consumption of the system. Through these contributions we provide a more thorough understanding of how HPC systems evolve power in their necessary compression stage. Overall our model enables us to lower power consumption $\sim$ 15% with a time tradeoff of 5% by scaling CPU frequency below base clock.

Additional Key Words and Phrases: lossy compression, green computing, power consumption, exascale computing

ACM Reference Format:
Grant Wilkins and Jon C. Calhoun (Advisor). 2021. Modeling Power Usage for the SZ Lossy Compressor on HPC Systems. 1, 1 (January 2021), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION AND MOTIVATION

As scientific computing and simulations become more complex, the amount of data produced and the energy usage of computing systems increases. As workloads approach exascale, compressing vast amounts of data ensures low overhead in data movement. Exascale computing is limited by its exorbitant power budget on the order of 20 MWs [5]. Lossy compressors — e.g., SZ [8] and ZFP [9] — have risen in popularity due to their ability to compress data more space efficiently with user-defined accuracy levels than traditional lossless compression. As lossy compressors become more prevalent and the size of data needing to be compressed increases, energy and time will need to be saved in data I/O using compressed data as found here [2], therefore understanding power usage of lossy compressors leads to more control over the power processes. Our contributions are outlined as follows:

- We characterize power usage and energy consumption of the SZ lossy compressor.
- We present a framework for modelling and optimizing based on CPU frequency.
- We test the hypothesis that this model enables optimal energy consumption *ab initio* for the given CPU and datasets.

Authors' address: Grant Wilkins, gfwilki@clemson.edu; Jon C. Calhoun (Advisor), jonccal@clemson.edu, Holcombe Department of Electrical and Computer Engineering, Clemson University, 433 Calhoun Drive, Clemson, South Carolina, 29634.

Manuscript submitted to ACM

2 METHODOLOGY

Three datasets from the SDRBench library [10] were utilized for compression and decompression. The outline of these datasets is included in Table 1 below. All fields for the datasets were considered, except for CESM-ATM which used six fields consistent with [3] and for symmetry with the other applications.

Application	Dimensions	# Fields	Size of Fields	Fields Considered
CESM-ATM [7]	$26 \times 1800 \times 3600$	13	673.9MB	CLOUD, CMFDT, ICLDTWP, QC, T, VU
HACC [6]	1×280953867	6	1046.9MB	$v_x, v_y, v_z, xx, yy, zz$
NYX [1]	$512 \times 512 \times 512$	6	536.9MB	b_dens, dm_dens, temp, v_x, v_y, v_z

Table 1. Datasets Considered in Study

We use the SZ (v2.1.8.3) [8] lossy compressor for compressing and decompressing the datasets. The HPC system used for these tests was a single Intel Xeon Silver 4114 Skylake CPU using Ubuntu 18.04 through CloudLab Computing [4]. Future work will include testing other lossy compressors and CPU models to increase the ubiquity of the energy model, these results are not available at this time due to time and resource constraints.

Perf, a tool for low-level performance monitoring, recorded the total energy consumption and runtime of SZ compression and decompression. We conducted experiments where a PSNR (peak signal to noise ratio, i.e. error-bound) value was selected as 30, 60, 90, 120, then SZ compression or decompression were run 100 times per CPU frequency step while the subsequent runtime and total energy were recorded. CPU frequency ranged from 0.8GHz to 2.2GHz at a step of 50 MHz. CPU frequency was set using cpufreq-set, a Linux tool to change CPU frequency. This was done for the datasets in Table 1.

The experimental design aimed to monitor power usage of SZ compression and decompression over CPU frequency, error bound, and different datasets. Several cycles of compression and decompression were run to isolate the process from OS background noise, data I/O and other unaccounted for variables in the power consumption.

3 RESULTS AND PURPOSE

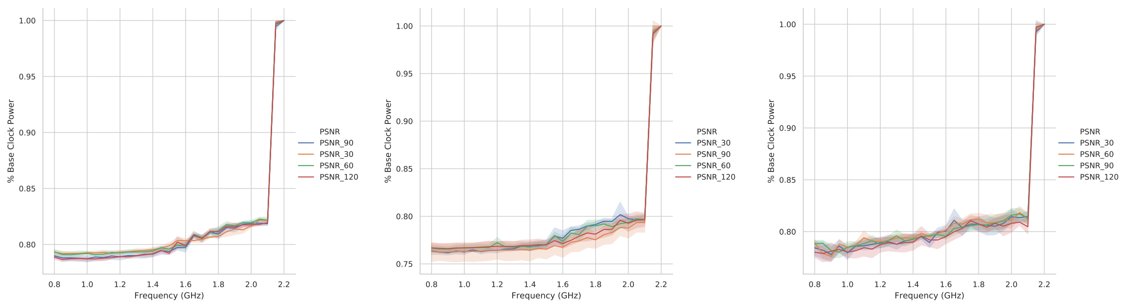
3.1 CPU Usage Results

For the experiments described in Section 2, the average power was calculated using the total energy and runtime data for SZ compression and decompression. Power is a more useful metric, as it is a measure that reflects energy through an expected runtime, via the relationship $P_{avg} = E/t$.

We scaled average power, dividing by P_{base} at $f = 2.2GHz$, giving a metric comparable across nodes. Figure 1 shows the power curves across the four PSNR values for decompressing the HACC, CESM, and NYX datasets.

The curves in Figure 1 reveal that power was consistent across error bound and showed the same trend between three different datasets. Error bound is an important characteristic of lossy compressors, but when looking at a black-box function of just frequency and base power, there were no discernible trends. While runtime increases with error bound, this does not mean greater power consumption, just greater total energy usage. This point confirms that a black-box model that computes power from CPU frequency is still useful in predicting SZ power usage.

From this we discerned that consistently one could lower the frequency 0.1GHz from the base clock and save 15% of power consumption with a 5% time tradeoff, on average. Future work aims to explain the spike at 2.1GHz as a software or hardware phenomenon.



(a) HACC (b) CESM-ATM (c) NYX

Fig. 1. Power characteristics of HPC system decompressing HACC, CESM, and NYX data

3.2 Early-Stage Model of SZ Power Consumption

Considering the power consumption of SZ as only as a black-box function $P_{SZ}(f, P_{base})$ that takes in a frequency and base power allows for a simple modeling task. When selecting the model to use, several models were curve fit as shown in Table 2. The model with the lowest root-mean squared error (RMSE) and sum of squared error (SSE) was chosen.

Model	P_{SZ} Fit	RMSE	SSE
Linear	$P_{base}(af + b)$	0.0451	8.488
Quadratic	$P_{base}(af^2 + bf + c)$	0.03694	5.694
Cubic	$P_{base}(af^3 + bf^2 + cf + d)$	0.03062	3.911
Exponential	$P_{base}(ae^{bf})$	0.04475	8.36
Power	$P_{base}(af^b + c)$	0.0229	2.188

Table 2. Curve fitting results of SZ average power using frequency

Taking our experimental results into account we posit the most adequate black-box model is a function $P_{SZ}(f, P_{base})$ where f is CPU frequency and P_{base} is the constant base clock power. The model of best fit, determined from the experimentally gathered power characteristics shown in Figure 1, is in the form:

$$P_{SZ}(f, P_{base}) = P_{base}(af^b + c)$$

for the experimentally determined a, b, and c in Table 3.

Fit Parameter	Fit Value (95% Confidence Interval)
a	2.237×10^{-9} $(1.027 \times 10^{-9}, 3.447 \times 10^{-9})$
b	23.44 (22.74, 24.13)
c	0.7841 (0.7833, 0.7849)

Table 3. Paraneters of curve-fitting best fit equation

The model utilized the compression and decompression results of all error bounds from the three datasets in Table 1, as there wasn't a significant difference in these variables while observing frequency. The goodness of fit statistics for the model are given in Table 4.

We found that the model was most accurate in the fairly constant region from 0.8GHz to 1.8GHz as shown in Figure 1, yet it still confirmed the assertion from section 3.1 of greater power efficiency below 2.1GHz. Considering that the

Metric	Power Model Statistic
R^2	0.8296
Adjusted R^2	0.8295
SSE	2.188
RMSE	0.0229

Table 4. Goodness of fit statistics for the presented power model, determined with least-squares fitting with 95% confidence

data varied from 0-1, the RMSE value informs that there is noticeable error in the model, yet based on the relatively low SSE and RMSE values in Table 2, this is still a fairly well fit model for an unrefined black-box approach.

The importance of a general model is to be able to *ab initio* scale a system for optimal power consumption during lossy compression. Future work will look at refining this model, yet this still models the power of an SZ compressor on a single-core architecture within reasonable statistical acceptability.

4 CONCLUSIONS AND FUTURE WORK

In this study we presented a description of power usage in HPC systems of SZ compression and decompression for several datasets. From this we designed a model for determining a power profile *ab initio*. The largest initial finding beyond these tools was that lowering frequency 5% decreased power usage by 15% on average with a trivial time cost. Future studies will aim to increase the number of datasets, compressors, and processors tested on to increase the ubiquity of the model and results. As exascale computing evolves, it is important that the power cost of compression is understood and optimized as lossy compressors will only increase in prevalence due to their assistance in data I/O..

REFERENCES

[1] Ann S. Almgren, John B. Bell, Mike J. Lijewski, Zarija Lukić, and Ethan Van Andel. 2013. Nyx: A MASSIVELY PARALLEL AMR CODE FOR COMPUTATIONAL COSMOLOGY. *The Astrophysical Journal* 765, 1 (feb 2013), 39. https://doi.org/10.1088/0004-637x/765/1/39

[2] Kenneth C. Barr and Krste Asanović. 2006. Energy-Aware Lossless Data Compression. *ACM Trans. Comput. Syst.* 24, 3 (Aug. 2006), 250–291. https://doi.org/10.1145/1151690.1151692

[3] S. Di, D. Tao, X. Liang, and F. Cappello. 2019. Efficient Lossy Compression for Scientific Data Based on Pointwise Relative Error Bound. *IEEE Transactions on Parallel and Distributed Systems* 30, 2 (2019), 331–345.

[4] Dmitry Duplyakin, Robert Ricci, Aleksander Maricq, Gary Wong, Jonathon Duerig, Eric Eide, Leigh Stoller, Mike Hibler, David Johnson, Kirk Webb, Aditya Akella, Kuangching Wang, Glenn Ricart, Larry Landweber, Chip Elliott, Michael Zink, Emmanuel Cecchet, Snigdhaswin Kar, and Prabodh Mishra. 2019. The Design and Operation of CloudLab. In *Proceedings of the USENIX Annual Technical Conference (ATC)*. 1–14. https://www.flux.utah.edu/paper/duplyakin-atc19

[5] Daniel Ellsworth, Tapasya Patki, Martin Schulz, Barry Rountree, and Allen Malony. 2016. A Unified Platform for Exploring Power Management Strategies. In *Proceedings of the 4th International Workshop on Energy Efficient Supercomputing* (Salt Lake City, Utah) *(E2SC '16)*. IEEE Press, 24–30.

[6] Salman Habib, Vitali Morozov, Nicholas Frontiere, Hal Finkel, Adrian Pope, Katrin Heitmann, Kalyan Kumaran, Venkatram Vishwanath, Tom Peterka, Joe Insley, et al. 2016. HACC: extreme scaling and performance across diverse architectures. *Commun. ACM* 60, 1 (2016), 97–104.

[7] JE Kay, C Deser, A Phillips, A Mai, C Hannay, G Strand, JM Arblaster, SC Bates, G Danabasoglu, J Edwards, et al. 2015. The Community Earth System Model (CESM), large ensemble project: A community resource for studying climate change in the presence of internal climate variability. *Bulletin of the American Meteorological Society* 96, 8 (2015), 1333–1349.

[8] X. Liang, S. Di, D. Tao, S. Li, S. Li, H. Guo, Z. Chen, and F. Cappello. 2018. Error-Controlled Lossy Compression Optimized for High Compression Ratios of Scientific Datasets. In *2018 IEEE International Conference on Big Data (Big Data)*. 438–447.

[9] P. Lindstrom. 2014. Fixed-Rate Compressed Floating-Point Arrays. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2674–2683.

[10] Scientific Data Reduction Benchmark. [n.d.]. https://sdrbench.github.io/. Online.