Integrating TEZIP into LibPressio: A Case Study of Integrating a Dynamic Application into a Static C Environment

ISITA TALUKDAR, University of California Berkeley, USA
AMARJIT SINGH (ADVISOR), RIKEN Center for Computational Science, Japan
ROBERT UNDERWOOD (ADVISOR), Argonne National Laboratory, USA
KENTO SATO (ADVISOR), RIKEN Center for Computational Science, Japan
WEIKUAN YU (ADVISOR), Florida State University, USA

CCS Concepts: • **Computing methodologies → Image compression**.

Additional Key Words and Phrases: high performance computing, neural networks, data compression, software integration

ACM Reference Format:
Isita Talukdar, Amarjit Singh (Advisor), Robert Underwood (Advisor), Kento Sato (Advisor), and Weikuan Yu (Advisor). 2023. Integrating TEZIP into LibPressio: A Case Study of Integrating a Dynamic Application into a Static C Environment. In . ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

SPring-8, LCLS-II-HE, SNS, and other instruments use software written in C and C++, producing huge volumes of time evolving data at high rate. [1] [8] [7] TEZIP is a neural network (NN) based compressor designed for high-quality compression of time-evolving data, but TEZIP is written in Python and is not easily usable from or ported to C++. Other compressors face similar issues, such as the LinLogCompress.jl in Julia [2] and compressors using PyTorch/TensorFlow, e.g., Autoencoder Based Compressor [4]. In this work, we develop new components in LibPressio that allow us to integrate with TEZIP and other external compressors efficiently and evaluate them with a systematic approach.

Integrating TEZIP and LibPressio requires building a bridge between python and C++ environments beyond what is offered by PyBind11, while also prioritizing efficiency for high performance computing.

2 BACKGROUND

TEZIP is an AI-based compressor designed for time-evolving data [6]. TEZIP predicts the next frame of data based on the previous, and stores the difference between the prediction and the true next frame for compression. LibPressio is an abstraction across compressors for dense tensors.

3 METHODOLOGY

Before TEZIP is used for compression, the image data must be used to train and test the PredNet model [5] that TEZIP uses for the aforementioned prediction. Image data is preprocessed into python objects for training and validation via

Manuscript submitted to ACM

hickle, which serializes to HDF5. Model architecture is saved in JSON format, along with the model weights in an HDF5 file. (See Figure 2 on Poster)

Unlike SZ3, TEZIP needs a training phase to train the Neural Network, where as SZ3's Lorenzo predictor does not require training [3], leading to overhead costs. Timing profiles of TEZIP when the compression command, as displayed below show opportunities for optimization during loading of the model and during prediction phase. (See Figure 3 on Poster)

For this study, we use SZ3, ZFP, and TTHRESH via LibPressio, and focus on the compression ratio metric and an artifact analysis, to analyze compression quality. LibPressio requires binary data as an input, while TEZIP must take a folder of RGB images. We developed a python script to scale the binary data into RGB images. Because of this conversion, only 3D datasets could be selected. We used three different 3D datasets from the SDRBench: Hurricane Isabel, NYX, and EXAFEL [9].

4 RESULTS

TEZIP compression ratio is the highest for all datasets at an absolute error bound of 1e-06. All compression was done with an absolute error bound. TEZip gets a much higher compression ratio (better) there are some artifacts generated during compression. There is a well-known tradeoff between quality and compressed data size in lossy compression. TEZIP's neural network operate by recognizing patterns in data, and is capable of being a "universal function approximator" in a way that the predictors/transforms in SZ3, tthresh, and ZFP are not. (See Figure 4 on Poster)

Compression of data can result in unexpected artifacts in the data, as demonstrated in comparisons of original versus decompressed slices of the Hurricane Isabel Data. TEZIP's neural network can erroneously shadow images' most probable input for a given space (Figure 5A, 5B). Similarly, in SZ3, interpolation error can result in one outlying data point creating a shadowing affect on the surrounding data (Figure 5C). Though the SZ3 artifacts are shown at an error bound of 1e-02, the same artifacts exist at smaller error bounds, but their breadth and magnitude get smaller as the bound gets smaller. (See Figure 5 on Poster)

5 CONCLUSION

Metrics can be generated for TEZIP compression and decompression via LibPressio. TEZIP compression ratios are higher than all other compressors (Figure 4). TEZIP's compression ratio (Error Bound 1e-06) for Hurricane Isabel is 128 which is 2.4 times greater than the leading SZ3's, 52.8.

In the future, we must optimize integration to enable fair comparisons to other compressors. We could utilize Mochi to avoid start up overhead. Additionally, we could utilize shared Memory to avoid copies and file sharing overheads. TEZIP's model training time could also be explored, as it is currently done independent of the LibPressio call to TEZIP.

In conclusion, this work sets a precedent for the integration of non C/C++ compressors into LibPressio. A similar framework can be used for other compressors in the future, leading to new developments in the field of data compression.

ACKNOWLEDGMENTS

Full Acknowledgments on the Poster

Supported by NSF award 1952302, ANL LDRD DE-AC02-06CH1135, COE research grant in computational science from Hyogo Prefecture and Kobe City through Foundation for Computational Science

REFERENCES

[1] Yasumasa Joti, Kyo Nakajima, Takashi Kameshima, Mitsuhiro Yamaga, Toshinori Abe, Kensuke Okada, Takashi Sugimoto, Takaki Hatsui, and Makina Yabashi. 2017. Data Analysis Environment for X-ray Free-Electron Laser Experiments at SACLA. *Synchrotron Radiation News* 30, 1 (Jan. 2017), 16–21. https://doi.org/10.1080/08940886.2017.1267556

[2] Milan Klöwer, M. Razinger, Juan Dominguez, Peter Düben, and Tim Palmer. 2021. Compressing atmospheric data into its real information content. *Nature Computational Science* 1 (11 2021), 713–724. https://doi.org/10.1038/s43588-021-00156-2

[3] Xin Liang, Kai Zhao, Sheng Di, Sihuan Li, Robert Underwood, Ali M. Gok, Jiannan Tian, Junjing Deng, Jon C. Calhoun, Dingwen Tao, Zizhong Chen, and Franck Cappello. 2023. SZ3: A Modular Framework for Composing Prediction-Based Error-Bounded Lossy Compressors. *IEEE Transactions on Big Data* 9, 2 (2023), 485–498. https://doi.org/10.1109/TBDATA.2022.3201176

[4] J. Liu, S. Di, K. Zhao, S. Jin, D. Tao, X. Liang, Z. Chen, and F. Cappello. 2021. Exploring Autoencoder-based Error-bounded Compression for Scientific Data. In *2021 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE Computer Society, Los Alamitos, CA, USA, 294–306. https://doi.org/10.1109/Cluster48925.2021.00034

[5] William Lotter, Gabriel Kreiman, and David Cox. 2017. Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning. arXiv:1605.08104 [cs.LG]

[6] Rupak Roy, Kento Sato, Subhadeep Bhattachrya, Xingang Fang, Yasumasa Joti, Takaki Hatsui, Toshiyuki Nishiyama Hiraki, Jian Guo, and Weikuan Yu. 2021. Compression of Time Evolutionary Image Data through Predictive Deep Neural Networks. In *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. 41–50. https://doi.org/10.1109/CCGrid51090.2021.00014

[7] Galen Shipman, Stuart Campbell, David Dillow, Mathieu Doucet, Jim Kohl, Garrett Granroth, Ross Miller, Dale Stansberry, Thomas Proffen, and Russel Taylor. 2014. Accelerating Data Acquisition, Reduction, and Analysis at the Spallation Neutron Source. In *2014 IEEE 10th International Conference on E-Science*, Vol. 1. 223–230. https://doi.org/10.1109/eScience.2014.31

[8] Chun Hong Yoon and Thomas A. White. 2018. Climbing the Data Mountain: Processing of SFX Data. In *X-Ray Free Electron Lasers*, Sébastien Boutet, Petra Fromme, and Mark S. Hunter (Eds.). Springer International Publishing, Cham, 209–233. https://doi.org/10.1007/978-3-030-00551-1_7

[9] Kai Zhao, Sheng Di, Xin Lian, Sihuan Li, Dingwen Tao, Julie Bessac, Zizhong Chen, and Franck Cappello. 2020. SDRBench: Scientific Data Reduction Benchmark for Lossy Compressors. In *2020 IEEE International Conference on Big Data (Big Data)*. 2716–2724. https://doi.org/10.1109/BigData50022.2020.9378449