Delta-Rice: A HDF5 Compression Plugin optimized for Digitized Detector Data

D. G. Mathews [1,2,¶], **C. B. Crawford** [2], **S. Baeßler** [1,3], **N. Birge** [4], **L. J. Broussard** [1], **F. Gonzalez** [1], **L. Hayen** [5,6,7], **A. Jezghani** [2,8], **H. Li** [3], **R. Mammei** [9,10], **A. Mendelsohn** [9], **G. Randall** [11], **G. V. Riley** [12], and **D. C. Schaper** [2,12]

1 Oak Ridge National Laboratory, Oak Ridge, TN, USA **2** Department of Physics and Astronomy, University of Kentucky, Lexington, KY, USA **3** Department of Physics, University of Virginia, Charlottesville, VA, USA **4** Department of Physics, University of Tennessee, Knoxville, TN, USA **5** Department of Physics, North Carolina State University, Raleigh, NC, USA **6** Triangle Universities Nuclear Laboratory, Durham, NC, USA **7** Normandie University, Rouen, France **8** Georgia Institute of Technology, Atlanta, GA, USA **9** University of Manitoba, Winnipeg, CA **10** University of Winnipeg, Winnipeg, CA **11** Arizona State University, Tempe, AZ, USA **12** Los Alamos National Laboratory, Los Alamos, NM, USA **¶** Corresponding author

Summary

Delta-Rice is an HDF5 (The HDF Group et al., 2020) filter plugin that was developed to compress digitized detector signals recorded by the Nab experiment (Fry et al., 2019), a fundamental neutron physics experiment. This is a two-step process where incoming data is passed through a pre-processing filter and then compressed with Rice coding. A routine for determining the optimal pre-processing filter for a dataset is provided along with an example GPU deployment. When applied to data collected by the Nab data acquisition system, this method produced output files 29% their initial size, and was able to do so with an average read/write throughput in excess of 2 GB/s on a single CPU. Compared to the widely used Gzip compression routine, Delta-Rice reduces the file size by 33% more with over an order of magnitude increase in read/write throughput. Delta-Rice is available on CPU to users through the HDF5 library.

Statement of Need

Many modern nuclear physics experiments, such as the Nab experiment, will produce petabytes of data. The cost and complexity of storing such a datasets motivated the development of a compression routine tailored specifically to the type of signals commonly recorded in these experiments. In these experiments, any compression routine must be fast enough to support real-time compression while also being lossless to prevent any reduction in the precision of offline analysis. Additionally, any candidate routine must be easily accessible to the various members of the collaboration and should not restrict users to a particular programming language to allow for a variety of analysis methods. 'Delta-Rice' was designed to meet these requirements and was implemented as an HDF5 filter plugin to ensure that each user can easily access data with minimal additional requirements in multiple programming languages (Mathews, 2022). While many other filter plugins exist for HDF5 files, such as Bitshuffle (Masui et al., 2015) and Gzip, Delta-Rice offers improved throughput and reduction in data size for many experimental efforts such as the Nab, NOPTREX (Schaper et al., 2020), and nEDM@SNS (Ahmed et al., 2019) efforts.

Algorithm Overview

This algorithm is a two-step process: the digitized signal is first passed through an encoding operation, such as delta encoding, to de-correlate the data and prepare it for the second step of Rice coding (Rice & Plaunt, 1971). These methods were chosen for this compression routine specifically for their simplicity, throughput, and storage efficiency. They also do not require a significant amount of additional information to be stored alongside the compressed data in order for the decompression routine to function, which improves storage efficiency further.

Rice Coding

Rice coding functions by encoding a value x in 2 pieces: q, the result of a division by a tunable parameter m, and r, the remainder of that division. q is stored in Unary coding, with r in truncated binary. In this routine, signed values are handled by interleaving positive and negative values as follows: $x' = 2 * x$ for $x >= 0$ and $x' = 2|x| - 1$ for $x < 0$. Rice coding is used instead of the more general Golomb coding (Golomb, 1966) because the restriction to powers of 2 for m allows for more efficient calculations. For information about the optimization of m, see Optimization. In the case that $q >= 8$, the output will be $q = 8$ followed by the original number in 16-bit signed representation. This is done to ensure that the amount a value can fail to be compressed is fixed. The outputs from this method are packed sequentially into 32 bit containers ensuring that no bits are wasted for any containers but the last one for a dataset.

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Initial Container	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1st Pack	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
2nd Pack	1	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0
After Output	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0

Figure 1: A demonstration of rice coding and bit packing when writing $x = -2$ and $x = 25$ with $m = 8$ for a 8 bit output container with a 16 bit temporary cache. Any remaining data in the temporary buffer is retained for the next write of x, or output at the end of the compression when no more values of x are provided.

Preparatory Encoding

Preparatory encoding is done to adjust the dataset to a form more optimal for Rice Coding. By default, this is done with delta encoding which stores the difference between subsequent values. The image below shows an example of this when applied to a signal from the Nab experiment. A simple optimization routine for determining the ideal filter is discussed in Optimization.
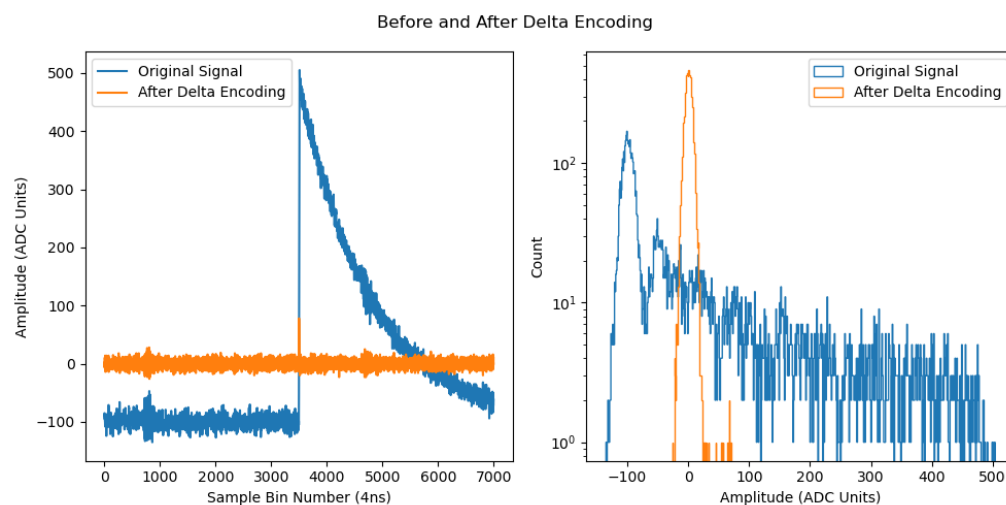
Figure 2: Left: A waveform before and after delta encoding. Applying Rice coding with $m = 8$ on the original signal expands the size of the waveform from 14 kB to 18.2 kB. The same Rice coding operation on the delta encoded waveform compresses the waveform to 4.6 kB, 33% the original size. Right: A histogram of a sample dataset before and after delta encoding. Note the clear reduction in the distribution width and that the most probable values are centered around 0.

Implementation

Delta-Rice is accessible to users through the HDF5 library (The HDF Group et al., 2020) as filter ID 32025. The user can specify m, the encoding filter, and the length of the smallest axis of the data being stored l. If l is specified and OpenMP (Dagum & Menon, 1998) is available, then the algorithm will utilize multiple threads to compress/decompress the data. Note that datasets written in parallel can be read by either serial or parallel decoding operations, but a dataset written serially will be read serially unless l was specified. For performance information and a discussion of using this routine on GPUs and FPGAs, see Performance.

Acknowledgements

References

Ahmed, M. W., Alarcon, R., Aleksandrova, A., Baeßler, S., Barron-Palos, L., Bartoszek, L. M., Beck, D. H., Behzadipour, M., Berkutov, I., Bessuille, J., Blatnik, M., Broering, M., Broussard, L. J., Busch, M., Carr, R., Cianciolo, V., Clayton, S. M., Cooper, M. D., Crawford, C., … Young, A. R. (2019). A new cryogenic apparatus to search for the neutron electric dipole moment. *Journal of Instrumentation*, *14*(11), P11017. https://doi.org/10.1088/1748-0221/14/11/P11017

Dagum, L., & Menon, R. (1998). OpenMP: An industry standard API for shared-memory programming. *Computational Science & Engineering, IEEE*, *5*(1), 46–55. https://doi.org/10.1109/99.660313

Fry, J., Alarcon, R., Baeßler, S., Balascuta, S., Palos, L. B., Bailey, T., Bass, K., Birge, N., Blose, A., Borissenko, D., Bowman, J. D., Broussard, L. J., Bryant, A. T., Byrne, J., Calarco, J. R., Caylor, J., Chang, K., Chupp, T., Cianciolo, T. V., … Zeck, B. (2019). The Nab experiment: A precision measurement of unpolarized neutron beta decay. *EPJ Web of Conferences*, *219*, 04002. https://doi.org/10.1051/epjconf/201921904002

Golomb, S. (1966). Run-length encodings (corresp.). *IEEE Transactions on Information Theory*, *12*(3), 399–401. https://doi.org/10.1109/TIT.1966.1053907

Masui, K., Amiri, M., Connor, L., Deng, M., Fandino, M., Höfer, C., Halpern, M., Hanna, D., Hincks, A. D., Hinshaw, G., Parra, J. M., Newburgh, L. B., Shaw, J. R., & Vanderlinde, K. (2015). A compression scheme for radio data in high performance computing. *Astronomy and Computing*, *12*, 181–190. https://doi.org/10.1016/j.ascom.2015.07.002

Mathews, D. (2022). *High performance data acquisition and analysis routines for the nab experiment* [PhD thesis, University of Kentucky]. https://doi.org/10.13023/etd.2022.446

Rice, R., & Plaunt, J. (1971). Adaptive variable-length coding for efficient compression of spacecraft television data. *IEEE Transactions on Communication Technology*, *19*(6), 889–897. https://doi.org/10.1109/TCOM.1971.1090789

Schaper, D. C., Auton, C., Barrón-Palos, L., Borrego, M., Chavez, A., Cole, L., Crawford, C. B., Curole, J., Dhahri, H., Dickerson, K. A., Doskow, J., Fox, W., Gervais, M. H., Goodson, B. M., Knickerbocker, K., Jiang, C., King, P. M., Lu, H., Mocko, M., … Visser, G. (2020). A modular apparatus for use in high-precision measurements of parity violation in polarized eV neutron transmission. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, *969*, 163961. https://doi.org/10.1016/j.nima.2020.163961

The HDF Group, N., Koziol, Q., & Science, U. O. of. (2020). *HDF5-version 1.12.0*. https://doi.org/10.11578/dc.20180330.1