# Efficient representation of sparse 3D dataset for the Milky Way

**Francesc Alted & Marta Iborra • The Blosc Development Team**

**contact@blosc.org • www.blosc.org**

## Abstract

Blosc2, a compression and format library, has introduced support for large multidimensional datasets. The extension includes special encoding of zeros for efficient handling of sparse datasets and a two-level data partition to reduce unnecessary decompression. The Blosc2 NDim layer allows for the creation and reading of n-dimensional datasets with fine-grained slicing and dicing. It optimizes compression by fitting different partitions to CPU cache levels. Blosc2 also incorporates Btune, a library that uses genetic algorithms and neural networks to quickly find the best compression parameters for a given dataset. This approach is much faster than traditional trial-and-error methods. The example provided demonstrates how Blosc2 NDim facilitates fast exploration of the Milky Way using the Gaia DR3 dataset.

## Introduction

Blosc is a high-performance compressor optimized for binary data, providing faster data transmission to the processor cache compared to traditional approaches. It utilizes the blocking technique and SIMD instructions to minimize memory bus activity and maximize compression/decompression speed.

Blosc2 is the latest version of the Blosc 1.x series, which is used in many important libraries, such as HDF5 [1], Zarr [2], and PyTables [3]. Its NDim feature excels at reading multi-dimensional slices, thanks to an innovative pineapple-style partitioning technique [4].

The example provided demonstrates the benefits of using Blosc compressed data for accelerating memory-bound computations, comparing its performance with other commonly used libraries.

## Methods

The new version of Blosc2 uses different techniques in order to be as efficient as possible. Next we will summarize some of these methods.

- **Zero suppression.** Blosc2 has the ability to check for zero values and suppress them making compression ratio bigger. This is significantly noticeable with sparse datasets like the Gaia dataset.
- **Two levels of partitioning**. Data is split into blocks (the smallest unit of data that can be compressed and decompressed independently) and chunks (a group of blocks that are compressed together).
- **Btune.** In Blosc you can choose from different compressors, filters, and other compression parameters. Btune is an AI tool which helps you choose those parameters depending on the tradeoff you specify between compression ratio and compression/decompression speed (see Table 1).

| Tradeoff | Most predicted | Cratio | Cspeed | Dspeed |
|---|---|---|---|---|
| 0.0 | blosclz-nofilter-5 | 786.51 | 106.86 | 91.04 |
| 0.1 | blosclz-nofilter-5 | 786.51 | 106.86 | 91.04 |
| 0.2 | blosclz-nofilter-5 | 786.51 | 106.86 | 91.04 |
| 0.3 | blosclz-nofilter-5 | 786.51 | 106.86 | 91.04 |
| 0.4 | blosclz-nofilter-5 | 786.51 | 106.86 | 91.04 |
| 0.5 | blosclz-nofilter-5 | 786.51 | 106.86 | 91.04 |
| 0.6 | zstd-nofilter-9 | 8959.6 | 8.79 | 59.13 |
| 0.7 | zstd-nofilter-9 | 8959.6 | 8.79 | 59.13 |
| 0.8 | zstd-nofilter-9 | 8959.6 | 8.79 | 59.13 |
| 0.9 | zstd-bitshuffle-9 | 10789.6 | 3.41 | 12.78 |
| 1.0 | zstd-bitshuffle-9 | 10789.6 | 3.41 | 12.78 |

**Table 1.** Btune prediction of the best compression parameters for decompression speed for the 3D Gaia array, depending on the tradeoff value between compression ratio (1) and decompression speed (0).

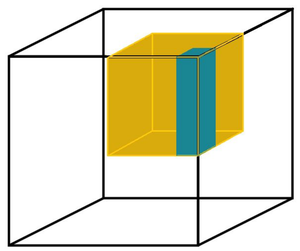

**Figure 1.** The blocks and chunks dimensions are parameters that can be tuned to fit the expected read access patterns.

## Results

In this study we create 4 arrays on disk containing the Gaia dataset, one for each different library version compared: Blosc2(NDArray), Zarr using Blosc1, H5py using Blosc1 and H5py using Blosc2. All have the same compression parameters: BloscLZ as compressor with no filters and compression level 5. According to Btune, we will be favoring decompression speed.

Next, we perform slices along an axis.

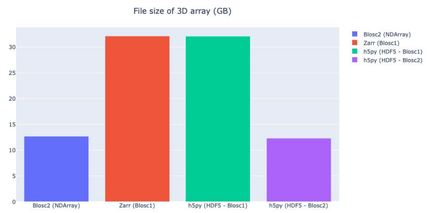


**Figure 2.** The libraries that use Blosc2 new version take advantage of the zero suppression. The compression ratio is doubled because of this.

## Results continued

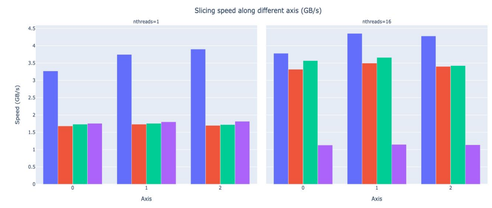


**Figure 3.** The Blosc2 library is way faster using only one thread, and its faster when using 16 threads (the optimal for our machine).

Finally, we did the same comparison but with the three different combinations that Btune predicted (see Table 1). We can check that Btune was not wrong at all!

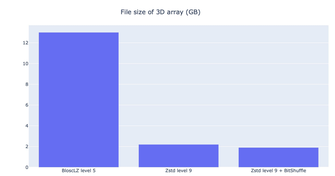


**Figure 4.** Sizes for each prediction.

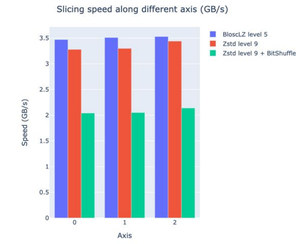


**Figure 5.** Slice speed for each prediction grouped by axis.

## Conclusions

With the new version of Blosc2 you can …

- Save the 7.3 TB Gaia dataset as up to only 2.1 GB using Zstd.
- Read slices of an array faster than any other compared library.
- Use Btune to easily choose which compression parameters fit best your needs.

## References

[1] The HDF Group, "Hierarchical Data Format, version 5," 1997-2023. www.hdfgroup.org/HDF5/.

[2] Zarr Developers, "An implementation of chunked, compressed, N-dimensional arrays for Python," 2017-2023. zarr.readthedocs.io.

[3] PyTables developers, "A Python package to manage extremely large data amounts of data," 2002-2023. www.pytables.org.

[4] Francesc Alted and Oscar Guiñón, "Introducing Blosc2 NDim", 2023.

## Acknowledgements

- Jordi Portell (Gaia Collaboration). For answering many questions about the Gaia dataset, and proposing possible explorations of it.
- NumFOCUS, a non-profit organization. They provided funds to the Blosc Development Team steadily through the past years.
- Huawei, a high-tech company, for an important and uninterested donation to the Blosc project.
- Sergio Barrachina (Universitat Jaume I). He has provided many advices, and code, during the development of the Btune project.