PROCEEDINGS OF SCIENCE

High Performance Computing algorithms for Imaging Atmospheric Cherenkov Telescopes

Thomas Vuillaume*
LAPP, Université Savoie Mont-Blanc, CNRS, France
E-mail: thomas.vuillaume@lapp.in2p3.fr

Pierre Aubert
LAPP, Université Savoie Mont-Blanc, CNRS, France
E-mail: pierre.aubert@lapp.in2p3.fr

Gilles Maurin
LAPP, Université Savoie Mont-Blanc, CNRS, France
E-mail: gillles.maurin@lapp.in2p3.fr

Jean Jacquemier
LAPP, Université Savoie Mont-Blanc, CNRS, France
E-mail: jean.jacquemier@lapp.in2p3.fr

Giovanni Lamanna
LAPP, Université Savoie Mont-Blanc, CNRS, France
E-mail: giovanni.lamanna@lapp.in2p3.fr

Nahid Emad
Laboratoire d'informatique Parallélisme Réseaux Algorithmes Distribués, UFR des Sciences, Versailles, France
E-mail: nahid.emad@uvsq.fr

PoS(ICRC2017)771

In the recent years, physics experiments have definitely entered the Big Data era. Future telescopes (such as the Square Kilometre Array, the Cherenkov Telescope Array, the Large Synoptic Survey Telescope, Virgo/LIGO...) will generate more data than ever. This is an exciting time as the analysis and the combination of all these data will lead to new discoveries and even to new ways to do science, but it comes with a price. Data management is now a challenge on its own. The H2020 project ASTERICS is addressing these challenges by bringing together the whole astrophysics and Astroparticle around the ESFRI projects to enable them to interoperate as an integrated, multi-wavelength and multi-messenger telescope. In particular, the OBELICS work package is developing common solutions for the generation, the integration and the analysis of data.

Under this framework, new solutions and algorithm based on high performance computing (HPC) to analyse data from Atmospheric Cherenkov Telescopes (IACT) are developed. Several developments are presented in this paper: 1. An original compression algorithm dedicated to integers. It is therefore especially interesting for physics experiments dealing with digitized signals such as IACT. Coupled with the LZMA algorithm, it considerably reduces the compression time while keeping a maximal compression ratio. 2. A HPC library with low level algorithms. Applied to the Hillas reconstruction using only reduced momentum and coupled with an adapted data format, these algorithms improve the computing times by a factor greater than 300. 3. A new reconstruction method based on Single Value Decompositions (SVD). This method compares the data to an image template (generated by Monte Carlo) using a handful of representative values. This considerably reduces the computation time and memory usage while extracting most of the information.

35th International Cosmic Ray Conference
10-20 July, 2017
Bexco, Busan, Korea

*Speaker.

1. Introduction

Astronomy and the new generation of instruments has entered the Big Data era. The generated volumes of data are unprecedented and this new paradigm imposes great constraints on architectures but also on data analysis algorithms.

The Cherenkov Telescope Array (CTA) is the next generation of Imaging Atmospheric Cherenkov Telescopes (IACT) and makes no exception. It will here serve as a use-case experiment. Thanks to its unprecedented sensitivity, it will greatly improve our understanding of the high-energy universe. But an improved sensitivity also means greater data volumes. The challenges in terms of data reduction and data analyses are immense and it seems that the most complex algorithms are hardly scalable. This will demand to investigate faster and more scalable algorithms.

This work describes a set of solutions developed to answer the challenges faced by upcoming physics experiments. The algorithms described here are very low-level and can be applied to a large variety of domains. They have been however tested and refined more specifically for IACT images.

In the first section, we present a new lossless compression algorithm for digitalized data. This algorithm has the advantage to provide a very good compression ratio while keeping very low compression and decompression times, thus being applicable to the reduction of huge data volumes.

In the second section, we present our high performance computing (HPC) library using vectorization solutions. The algorithms in this library are low-level but have been applied to the computation of Hillas parameters for IACT images. These parameters are at the core of most of the IACT data analysis algorithms.

In the last section, we propose an idea for a new reconstruction method based on spectral value decomposition. These values present the advantage to contain more information on the images than the Hillas parameters, thus allowing for more complex analysis including more physics.

The developments presented here has been realized in the framework of the ASTERICS-H2020 project and as such are available as an open-source software package to the Astronomy, Astrophysics and Astroparticle communities.

2. A lossless compression algorithm for digitalized data

The colossal volume of data generated by current and future physics experiment (not only in astronomy) impose challenges not only in the data analysis but also in the transfer and storage of these data. Increase in hardware sizes will not be sufficient to solve those challenges. However, improvements in the data compression can help addressing them. The most efficient compression algorithms are generally used for picture (JPEG), video (H264), music (MP3) files, allowing a compression ratio greater than 10. However these algorithms are lossy and cannot be used in the case of physical data where data loss is not acceptable. Lossless compression is therefore unavoidable. Characters lossless compression, CTW (Context Tree Weighting) [5], Burrows-Wheeler transform, LZ77 [7], LZW [6] or PPM [4], can not be used on physical data as they do not have the same characteristics (like characters occurrence or characters repetition) as the textual ones.

The most commonly used methods for loss compression on binary data are the Lempel-Ziv-Markov chain algorithm (LZMA)[1] developed by Lempel and Ziv (1996), LZ78 [8], BZIP [3],

GZIP[2] or the Huffman algorithms. They provide good compression ratio (up to 4.8 on binary data), however they are quite slow comparing to the data flow of the coming experiments.

Many experiments in physics acquire digitalized (12 bits for example) signals dominated by white noise in unsigned integers. In many examples, the white noise can be represented by a Gaussian distribution and often dominates the signal - generally a Poissonian distribution. In these cases, most of the data is noise and follow a Gaussian distribution. In many cases, the range of values $b = v_{max} - v_{min} + 1$ (with v_{min} the data minimum value and v_{max} the data maximal value) covered by the data is much smaller than the range of values covered by an unsigned int $d = 2^{32}$. It is therefore possible to store several data in the same unsigned int. By subtracting v_{min} to the data, b becomes its basis. Once this basis and the data minimum value stored, the data can be stored as contiguous basis blocks, with several blocks (at most d/b) in one unsigned int (see figure 1). The complete presentation of the compression algorithm is done in another article (submitted). As an example, an IACT simulation file such as the ones used in CTA (Simtel_array) of 1.2Go can be compressed with a ratio of about 3.75 in only 3.7 seconds whereas the use of the LZMA algorithm results in a compression ratio of 4.8 in more than 7 minutes.

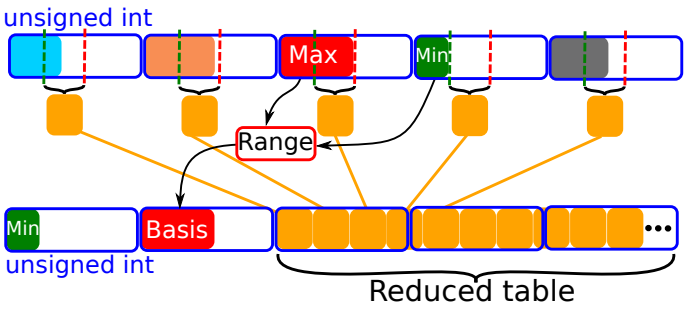

Figure 1: Illustration of the reduction principle. The upper line represents the data (different colours for different values). In the second line, the orange block represent the changes between the different values to compress. The last line shows the compressed data (as they are stored). First, the minimum value of the data, next, the basis $b = max - min + 1$, and finally the data stored as contiguous blocks. The main idea is to notice that several data can be stored in the same unsigned int and only the changes between the data are stored. The common parameters like the range of the data (minimum and maximum or compression basis) are stored only once.

3. Vectorization applied to Hillas parameters calculation

The typical image of an atmospheric shower by an IACT is can be approximated by an ellipse. The Hillas parameters are the parameters that best describe this ellipse (namely the barycentre, the first and second momentum of the signal) and can be computed analytically. Based on these parameters computed for each camera image of the IACT network, a stereoscopic reconstruction can give the direction and the energy of the primary photons. Improvement of the computing performances has been realized through two axes: modification of the data format and vectorization of the computation. By changing the data format from a root-based data format to an optimized data

Reduction	Speed cy/el	Acceleration
C/C++	2.69842	1
GCC, SSE4	0.72845	3.839281

Table 1: Speed and speed up for float reduction for basic C/C++ code and GCC vectorization on SSE4 architecture.

Interleaving	Speed cy/el	Acceleration
1	0.616841	4.374579
2	0.338947	7.961185
4	0.226990	11.887836
8	0.226675	11.904356

Table 2: Speed up for interleaved float accumulation on SSE4 architectures.

Interleaving	Speed cy/el	Acceleration
2	0.172954	15.601951
4	0.115568	23.349197
8	0.11379	23.714034

Table 3: Speed up for interleaved float accumulation on AVX architectures.

format with aligned data and allowing data prefetching to optimize the use of the CPU Cache-L1, we observed a speed up in the computation of 40 (see table 5).

The main optimization consist into the reduction optimization. More generally, the vectorization and optimization of a computation with back dependencies.

3.1 Reduction Optimization

The vector reduction is the basis of many algorithms. So, its optimization is crucial. Basically, a reduction written in C/C++ has a speed of 2.69842 cy/el, due to the architecture. The GCC vectorization on SSE4 architecture gives a speed up of 3.8 (see table 1).

The GCC compiler can not have better performances because the CPU pipeline is not full due to the reduction computation back dependency (the result of the previous addiction has to be known to compute the current one).

The back dependency can be avoided by using several accumulator instead of one. This interleaving able the CPU pipeline to compute a reduction part without needed the very previous result. The number of used accumulator has to be fine-tuned for each CPU architecture (see SSE4 in table 2, and AVX in table 3).

3.2 A fast barycentre calculation

The barycentre computation can be optimized the same way as the reduction (see in section 3.1). But, in this case, the computation time over a single element is greater than a reduction

momenta	Architecture	Speed cy/el
First 1D	SSE4	0.457595
	AVX	0.179133
First 2D	SSE4	0.910941
	AVX	0.343091
First and second 2D	SSE4	1.29088
	AVX	0.599513

Table 4: First and second momenta speed up example on SSE4 and AVX architecture.

because the computation is more complex (addiction and multiplication). So, less interleaving is needed.

The table 4 shows the speed up obtained by managing the vectorization by hand on the barycentre computation.

The vectorized reduction and barycentre calculation, are part of the open source PLIB_FAST library bring another important speed up (greater than 8 for SSE4 architectures and greater than 16 for AVX architectures). This result brings a very important total speed-up (see table 5).

	Speed (cy/el)	Speed up
Initial Hillas	2125.5	1
Home-made data format	53.1375	40.0
Vectorized Hillas SSE4	6.39931	332.14516
Vectorized Hillas AVX	2.98499	712.06268

Table 5: Comparison between the different optimizations perform on the Hillas parameters calculation.

4. More complex analysis based on singular value decomposition

Template analysis consists in comparing observation data with simulated data in order to retrieve the primary physical conditions that generated the signal. This approach has proved to improve physics performances. It is currently used for gravitational waves analysis [1] and for IACT data analysis [2]. However, complete templates may represent a lot of simulated data to look into. It requires consequent computing resources and time and for current and future experiments, may be unachievable. Here we propose to combine this approach with a data reduction using Singular Value Decomposition (SVD). SVD is a mathematical solution to extract spectral values from a rectangular matrix. SVD decomposition is a reversible process if the complete spectrum is kept. However, a couple of values is often enough to characterise the essential physics information. In this case, the treatment is much faster and template analysis can be used on the SVD spectrum.

Principle: SVD is applied to a set of images (simulated data) generating a singular value spectrum for each of them. A label (including the physical parameters) is associated to this spectrum. The generated spectra form a dictionary to which a new unknown image spectrum will be

compared. A likelihood can be computed to find the most probable templates and finally get the corresponding label (see right figure).

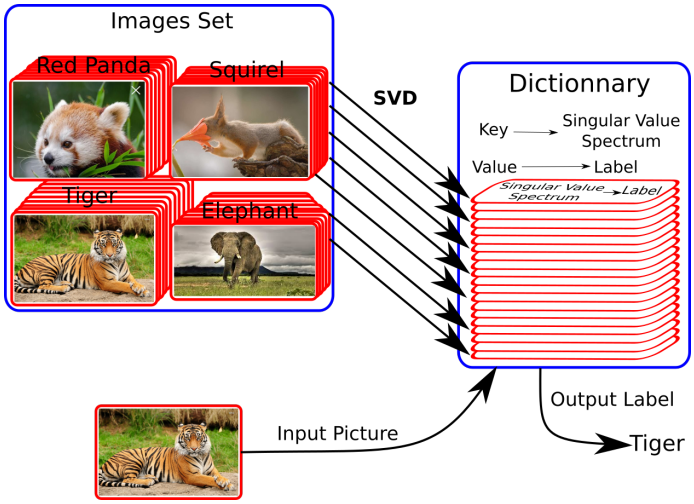

Figure 2: SVD principle: a dictionary of singular value spectra (keys) is generated from images associated to physical values (label). To analyse a new image, its spectrum is computed and compared to the bank of spectra using a likelihood function to find the most probable label.

The template analysis aims at reconstructing the detected shower by precomputing IACT answer with shower simulations and by comparing the real telescopes pictures to the simulated ones. They have good physical results on IACT data, particularly on discrimination.

Unfortunately, this pixel per pixel pictures comparison is very slow. There are a lot of pixels in the new IACT cameras and a single pixel comparison is time consuming. Moreover, the classical template method are not scalable to the new IACT data volume and structure. The pixel per pixel comparison must be replaced by a lighter method. The Singular Value Decomposition is one of these methods.

The Singular Value Decomposition algorithm (SVD) generalized the Fourier transform on matrices. The matrix SVD is :

$$M = UDV^T \tag{4.1}$$

Where $M \in M_{n,m}(\mathbb{R})$, $U \in O_{n,m}(\mathbb{R})$ (orthogonal matrices $n \times m$), $V \in O_m(\mathbb{R})$, $D \in D_m(\mathbb{R})$ (diagonal matrix). The diagonal matrix D is also called singular value spectrum. This spectrum condenses the matrix M information as the frequency spectrum in a Fourier transform.

Therefore, the singular value spectrum can be compared as the pixel per pixel comparison to get the primary particle physical characteristics. This method used less data because only few singular values (~ 7) are relevant to describe the signal in the camera. This method is also scalable compared to the classical IACT template analysis because all the camera pictures can be processed at the same time.

5. Conclusion

With the development of new science infrastructures generating unprecedented volumes of data, new HPC approaches are necessary in order to manage and analysis scientific data. In this work we present generic solutions including a new lossless compression algorithm for digitized data, a HPC library providing low-level algorithms and a new approach for template analysis. These developments have been tested and optimised for IACT data. They are available to download[1] as open-source algorithms and can therefore be used by the whole community.

We acknowledge support from the ASTERICS project supported by the European Commission Framework Programme Horizon 2020. Research and Innovation action under grant agreement n. 653477

References

[1] 7zip compression algorithm. www.7-zip.org.

[2] Gzip compression algorithm. www.gzip.org/index-f.html.

[3] M. Burrows and D. J. Wheeler. A block-sorting lossless data compression algorithm. Technical report, Systems Research Center, 1994.

[4] I. Clear and I. Witten. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Information Theory*, COM-32, 1984.

[5] Y.M. Shtarkov F.M.J. Willems and T.J. Tjalkens. The context-tree weighting method: basic properties. *IEEE Transactions on Information Theory*, 41, 2002.

[6] Jan Platos and Jiri Dvorský. Word-based text compression. *CoRR*, abs/0804.3680, 2008.

[7] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, May 1977.

[8] Jacob Ziv. A constrained-dictionary version of LZ78 asymptotically achieves the finite-state compressibility for any individual sequence. *CoRR*, abs/1409.1323, 2014.

[1] Download at https://gitlab.in2p3.fr/CTA-LAPP