hipeCTA

a high performance library for the Cherenkov Telescope Array data analysis

T. Vuillaume*, P. Aubert*, J. Jacquemier*, G. Maurin*, G. Lamanna*, for the CTA Consortium+

*Laboratoire d'Annecy de Physique des Particules, Université Grenoble Alpes, Université Savoie Mont Blanc, CNRS, LAPP, 74000 Annecy, France,
+ *See www.cta-observatory.org*

ABSTRACT

The Cherenkov Telescope Array (CTA) is the next-generation observatory for ground-based gamma-ray astronomy. CTA's baseline layout comprises two arrays of gamma-ray telescopes in both hemispheres, with 19 telescopes on the island of La Palma (Spain) and 99 telescopes in Paranal (Chile). Due to its large number of telescopes, CTA will record a tremendous amount of data (more than 3PB/year) that represents a computing challenge requiring a performant reconstruction software. We have developed a high-performance algorithm able to tackle these challenges and to perform the reconstruction of CTA raw data maximizing the usage of computing resources and thus minimizing their cost.

As Python is becoming the standard language in gamma-ray astronomy for data processing, we developed a Python library, hipeCTA, using wrapped optimized C++ code, thus ensuring efficiency, ease of use and integrability with other common libraries, especially ctapipe, the prototype library for CTA low-level data processing.

Here we present hipeCTA and show that the obtained physics and computing performances could allow a real-time analysis consistent with CTA requirements with reasonable computing resources.

Supported by:

The Challenge

Imaging Atmospheric Cherenkov Telescopes (IACT) data require a complex analysis to reduce sets of images to event parameters. In order to perform this event reconstruction in CTA, the ctapipe [1] prototype framework is under development using Python as main language.

The volume (several PB/year) and rate (several GB/s) of data produced by CTA is challenging developers to provide high performance algorithms that are easy to use by pipeline developers with a wide range of skills

hipeCTA

Python provides several advantages:

- **Modularity**
- **Ease-of-use**
- **Wide usage in astronomy (better acceptance, re-use and tool integration)**
- **Performance optimization thanks to scientific libraries (e.g. Numba or Numpy)**

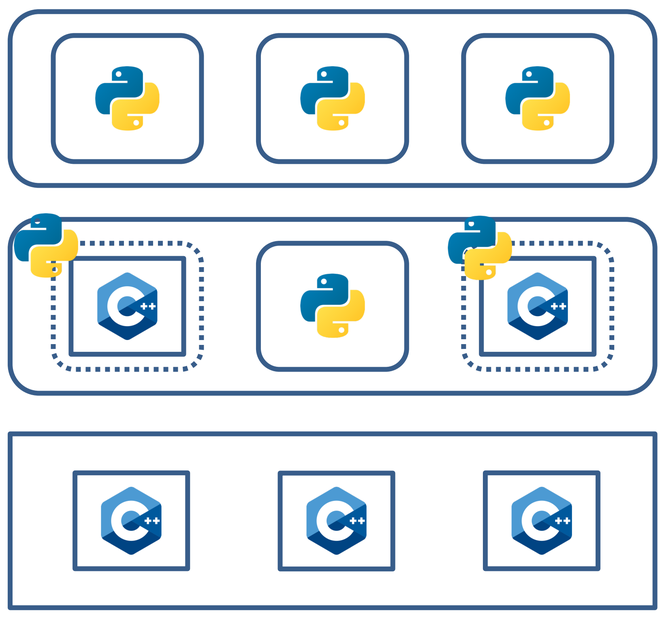
However, numba or numpy are generic libraries and can still be optimized for specific usage and complex cases.

hipeCTA [2] is a Python library providing highly optimized algorithms specific to IACT data analysis using modern CPUs capabilities such as vectorization and data prefetching. These algorithms are developed in C++ for performances and wrapped in Python for integration to ctapipe or other Python frameworks.

hipeCTA approach for CTA

Key algorithms existing in ctapipe are mirrored in hipeCTA, conserving the same API and of course giving the same results. This allows users transparent inter-change between ctapipe's algorithms and hipeCTA's ones.

This is very convenient for tests and to speed-up a specific part of the pipeline.
If required, for example for real-time analysis where best computing performances are needed, a complete C++ pipeline can be compiled.



1. **A pure Python pipeline (ctapipe)**
2. **Some algorithms are replaced by their C++ optimized version wrapped in Python. The pipeline is unchanged.**
3. **All algorithms are used in their C++ version. A C++ pipeline can be compiled to obtain the best performances.**

Advantages:

- **Best of both worlds: providing computing performances in an user-friendly, easy to integrate environment**
- **Result consistency: unit-test C++ algorithms versus their Python version**
- **API consistetncy: interchangeable algorithms between their Python and wrapped C++ version**

Results

Best performances are obtained with data formats allowing to take full advantage of the HPC algorithms. That is why we also developed a specific data format, hipeDATA, for simulated Monte-Carlo data, to compare with eventio [3], data block format developed specifically for iact data.

On DISK (SSD)	ctapipe	hipecta
eventio	130Hz (36MB/s)	196Hz (54MB/s)
hipeDATA	-	350Hz (98MB/s)

On RAM (DDR4)	ctapipe	hipecta
eventio	180Hz	370Hz
hipeDATA	-	6300Hz

Comparison of ctapipe (v0.6.2) and hipeCTA performances. The speed-up is particularly important in RAM, making hipeCTA C++ pipeline a good candidate for CTA real-time analysis.

ACKNOWLEDGEMENTS
We gratefully acknowledge financial support from the agencies and organizations listed here:
www.cta-observatory.org/consortium_acknowledgments
Funded by the European Union's Horizon 2020 – Grant N° 824064

References
[1] ctapipe: A Low-level Data Processing Framework for CTA, K. Kosack et al, 36th International Cosmic Ray Conference (ICRC2019), July 2019
[2] hipeCTA: https://gitlab.in2p3.fr/CTA-LAPP/HiPeCTA
[3] https://www.mpi-hd.mpg.de/hfm/~bernlohr/sim_telarray/Documentation/eventio_en.pdf