Asterics
Astronomy ESFRI & Research Infrastructure Cluster

ASTERICS - H2020 - 653477

Technology benchmark report – D-ANA

ASTERICS GA DELIVERABLE: D3.10

Document identifier:	ASTERICS-D3.10-final.docx
Date:	**04-05-2017**
Work Package:	**WP3 OBELICS**
Lead Partner:	**UCAM**
Document Status:	**Report**
Dissemination level:	**Public**
Document Link:	www.asterics2020.eu/documents/ASTERICS-D3.10.pdf

Abstract

In this report, we present the relatively diverse set of results and work that has been done and the work in progress under task 3.4 of OBELICS WP with focus on benchmarking and in

particular comparison and evaluation against existing systems. This is a sub-set of the work done since some of the tools being developed do not fit into the technology evaluation or benchmarking focus.

I. COPYRIGHT NOTICE

II. DELIVERY SLIP

	Name	Partner/WP	Date
From	Bojan Nikolic	UCAM	10/04/2017
Author(s)	• Bojan Nikolic, Peter Hague (UCAM) • Jean Jacquemier (LAPP), • Cristiano Bozza, T Chiarusi , R. Coniglione , A. Martini , P. Migliozzi , C. Pellegrino ,B. Spisso (INFN) • Liam Quinn (CPPM) • F. Pasian, S. Bertocco, A. Bignamini, A. Costa, C. Knapic, M. Molinaro, G. Taffoni (INAF) • E. Chassande-Mottin (APC).	• UCAM/WP3 • LAPP/WP3 • INFN/WP3 • CPPM/WP3 • INAF/WP3 • APC/WP3	10/04/2017
Reviewed by	Giovanni Lamanna	LAPP/WP3	20/04/2017

Approved by	AMST		03-05-2017

III. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
1	10/04/2017	First Draft	Bojan Nikolic (UCAM)
2	02/05/2017	Final Draft	Bojan Nikolic (UCAM)

IV. APPLICATON AREA

This document is a formal deliverable for the GA of the project, applicable to all members of the ASTERICS project, beneficiaries and third parties, as well as its collaborating projects.

V. TERMINOLOGY

ASTERICS	Astronomy ESFRI & Research Infrastructure Cluster
CTA	Cherenkov Telescope Array
ESFRI	European Strategy Forum on Research Infrastructures
KM3NeT	Cubic Kilometre Neutrino Telescope
OBELICS	Observatory E-environments LInked by common ChallengeS
SKA	The Square Kilometre Array

Table of Contents

1. Introduction

The ASTERICS task 3.4 of the work package OBELICS (Data ANAlysis/ interpretation, D-ANA) is focused on two main themes:

1. Tools to analyse and interpret astronomical/astro-particle observations in an efficient manner (both in the sense of statistical efficiency and computational efficiency)
2. Tools for controlling access to these observations in an appropriate way, allowing efficient remote and distributed analysis and working

In this report, we present the work that has been done in OBELICS with focus on benchmarking and in particular comparison and evaluation against existing systems. This is a sub-set of the work done since some of the tools being developed do not fit into the technology evaluation or benchmarking focus.

Benchmarking plays a critical role in the OBELICS workplan as already formulated at the proposal stage. The reason for this is that many of the challenges for the next generation observatories are challenges of scale and efficiency – can we process a much larger quantity of data; can we efficiently extract information from these larger quantities of data? The overall scientific output and efficiency of the observatories partnering in OBELICS will, to a large extent, be governed by these challenges of scale. For example, typical SKA datasets will be around 100 times larger than the largest comparable datasets today. To summarise, for these facilities efficiency (both in term of computing time and extracting information) becomes more important than simply functionality.

The purpose of benchmarking at this stage of the OBELICS project is both to identify areas where improvements in software will have a significant positive effect on the scientific capabilities future observatories and to allow selection from competing technologies. Since task 3.4 is only focused on open-source software there are no procurement implications of the benchmarking. The outputs will be used to guide to further work within the OBELICS project and to quantitatively measure the improvements made by the OBELICS project.

The choice of technologies to benchmark against has at this stage been driven in part by the existing technology choices in the individual facility projects and in part by survey of commonly used community packages. A full survey of all possibilities was not practical, and also of limited usefulness in some areas since major technology changes could be difficult, so that created an effective pre-selection.

2. Statistical analysis of multi-wavelength observations

We are developing a Markov Chain Monte Carlo (MCMC) process to detect and characterise sources in aperture synthesis images. This is based on already existing software in our department, which for the purposes of OBELICS will be expanded, scaled, documented and integrated with existing Python libraries. We are testing this against a baseline of established source detection methods such as SExtractor and Aegean. The process will also be able to incorporate evidence from observations at other wavelengths and from established source catalogues directly into the likelihood calculations being done in the data space.

For the purposes of development, we are using ALMA archive data, which has in most cases already been well studied with existing methods. Details of the processing of this archive can be found in our contribution to the 3.3 D-INT Technology Benchmark Report.

2.1 Software Implementation

Prior to the commencement project, statistically reliable MCMC source detection in a Bayesian framework was perhaps best done using BayeSys (www.inference.phy.cam.ac.uk/bayesys/) - a program which is complex and not widely used in the astronomical community. As part of this work package, we have recreated the likelihood calculator as a C++ class that can be interfaced with other MCMC software and have written a Python wrapper so that it can be invoked by PyMC. The same class can be used to perform an identical calculation in a C++ MCMC driver, RainfallMCMC (https://github.com/petehague/RainfallMCMC). PyMC is well known but it does not provide the functionality for multiple atoms. RainfallMCMC is another in-house code, simpler and more accessible than BayeSys, but not yet with the same level of functionality. However, it is easier to extend than BayeSys.

The MCMC driver used must be accessible through Python in order to allow for operation alongside common libraries such as astropy and numpy. PyMC has an advantage in this respect, but it also has a disadvantage in that it would be harder to maintain a parallel fork of PyMC that includes the functionality we require. Thus, we have chosen to push forward the development of RainfallMCMC to the point where it can support the required functionality.

2.2 Comparison with other methods

We use raw data and products from the ALMA archive for testing. Most projects archive CLEAN-ed images that have been corrected for primary beam sensitivity. In cases where we have been able to obtain the primary beam flux images, the correction has been undone for the purposes of source identification even though this results in substantially incorrect fluxes for sources that are not close to the pointing direction of the antennae.

SExtractor is an optical source detection method and must operate on an CLEAN-ed image that is the closest analogue to a CCD or scanned photographic plate. The CLEAN algorithm has to be considered a part of the source detection process here. We do not consider SExtractor to be a long-term solution, it is included here for comparison as a widely used source finding tool. Aegean also operates on Cleaned images, but is built with more awareness of the nature of interferometry products and thus had advantages in characterising sources.

In this example, we use an image of the centre of NGC 1808 for a comparison. SExtractor picked up more spurious sources than Aegean did, and there was also disagreement between the two methods in size and shape of the half maximum ellipses enclosing the sources. The MCMC process matched the sources found by Aegean well, but provided additional information regarding the shape of the extended main source - this is in fact the active central black hole of NGC 1808.

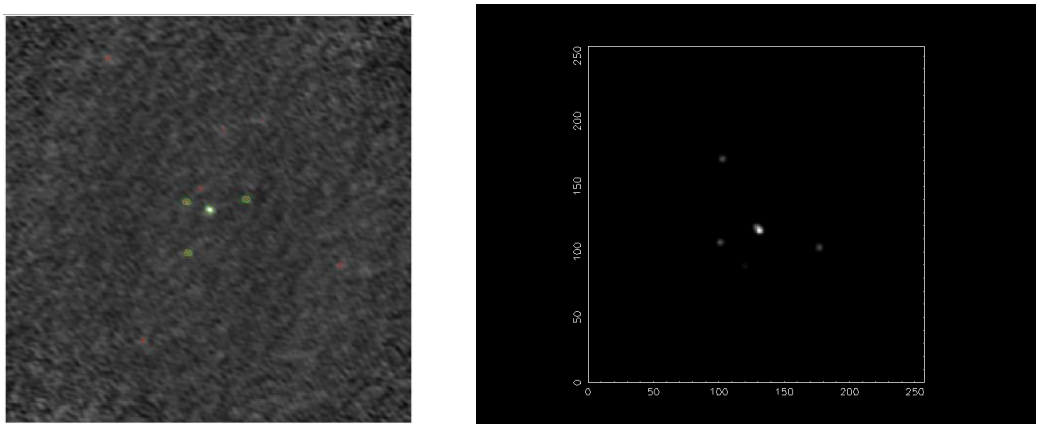

***Fig 1: Left** Half maximum ellipses found in the clean map of the NGC 1808 observations by Sextractor (red) and Aegean (green) and **right** Results from the MCMC process using multiple atoms. Note that the MCMC result is rotated 90 degrees relative to the other result.*

3.Likelihood-based methods for event reconstruction

We are developing a series of C++ classes and executables to identify and reconstruct incoming events in astroparticle physics experiments. These methods are created and tested within the context of the KM3NeT, building upon existing software and designed to be used either as a standalone or integrated with ROOT.

The ability to construct, store and access probability density functions (PDFs) is crucial when developing an effective reconstruction strategy. With this in mind, we have developed a tool which builds multi-dimensional data structures for PDF storage, taking Monte Carlo data as input. Once built, these structures can be purposed for event reconstruction or simply as a means to crosscheck between the Monte Carlo and PDFs calculated through another method.

Once created, these PDFs can be interfaced with our in-house implementations of the Nelder-Mead and Levenberg-Marquandt methods for function minimisation. Integration of existing standard minimisers, such as Minuit, is also possible with the creation of a simple wrapper function. Photon arrivals at photomulitiplier tubes (PMTs) are treated using either a poisson distribution or a continuous counterpart, for cases where hit arrival time is particularly important. It should be noted that, in the discrete case, empty PMTs can also be incorporated into the fit, yielding more information about the shape of the event.

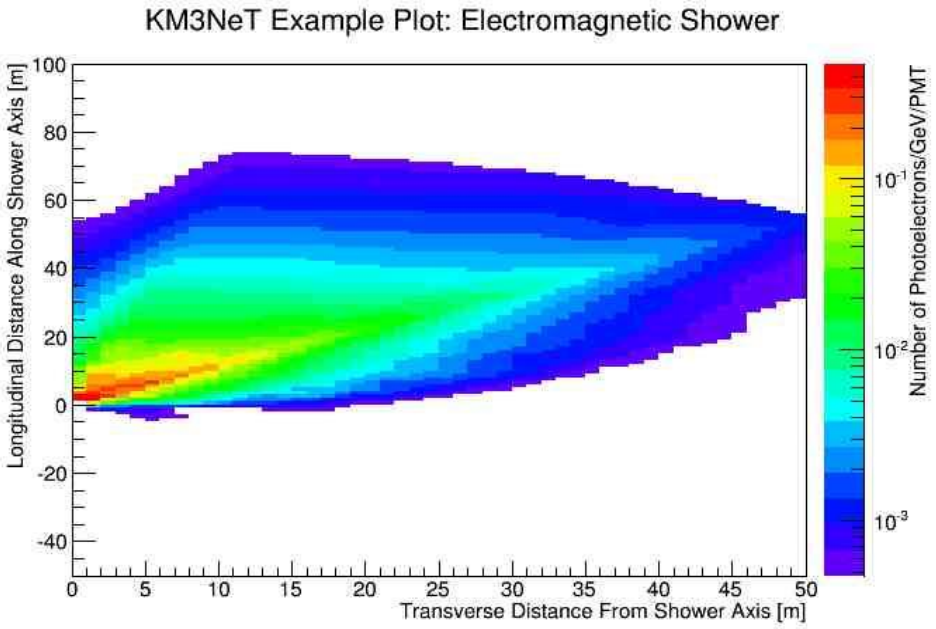

Fig 2: Example plot- A 2D projection of the probability density function of an electromagnetic shower in KM3NeT, for a photomultiplier tube facing in the opposite direction to the shower axis.

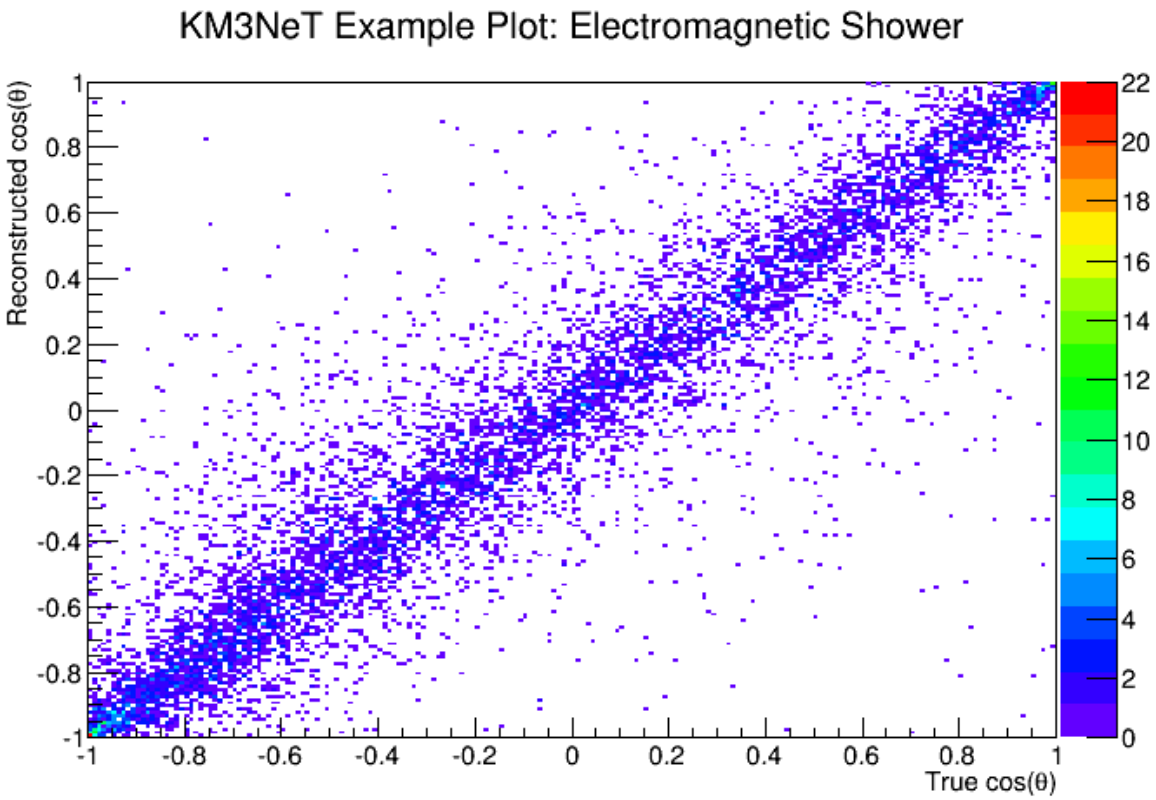

Fig 3: Example plot - Reconstruction output from electromagnetic showers, using poisson statistics and the Nelder-Mead method for minimisation. A clear, unbiased correlation between the true and reconstructed azimuth angle of the event is seen.

4. Authentication & Authorisation technology

This section summarizes the solutions examined within D-ANA to implement Authentication and Authorisation (A&A) mechanisms within pilot implementations for four projects: CTA, SKA, the INAF IA2 archives, and an IVOA-compliant implementation of interoperability between the EGI and CANFAR clouds for research. The latter pilot implements coordination of A&A activities between the ASTERICS work-packages OBELICS and DADI.

A complete report on the benchmarking activities on A&A technologies performed in the framework of D-ANA is available on the open ASTERICS wiki at "Authentication & Authorisation Technology Benchmarking Report" (draft 0.3, 04 April 2017).

As reference documentation, Implementation Notes are available for the Authentication & Authorisation pilots prepared up to now for the projects involved in ASTERICS (CTA, SKA, IVOA), plus a pilot performed for the INAF-IA2 archives.

- Implementation Note: SKA A&A (draft 0.2, 30 Mar 2017)
- Implementation Note: IVOA Users and Data Access Management (draft 0.2, 31 Mar 2017)
- An A&A best practices (draft 0.1, 30 Mar 2017) has been produced as the skeleton of a living document to be delivered in its final form at M48.

The work performed takes into consideration several suggestions coming from other EU H2020 projects such as Indigo-DataCloud and AARC, and make use of their deliverables. Both projects performed a detailed analysis of the technologies currently available, and their deliverables in both cases are lists of compliant technological stacks and freely available software tools that try to integrate several Authentication protocols.

The technologies used for the implementation of the CTA, SKA and IA2 pilots include:

- Shibboleth (used by the CTA and SKA pilots) is an Identity Provider (IdP) providing Single Sign-On services and extending reach into other organizations and new services through authentication of users and securely providing appropriate data to requesting services. Versions 3.x allow “consent management”, a feature that allows users to be prompted to consent to attribute release (this feature is specifically used for CTA). Deployed in the majority of federations, Shibboleth requires specialized expertise to operate: this issue can be mitigated by the availability of high quality documentation produced by the open source community.

- SAML (v2.0 - March 2005 - used by the CTA, SKA and IA2 pilots) is a common XML-based open-standard defining a framework for exchanging authentication and authorization data between parties, in particular, between an Identity Provider (IdP) and a Service Provider (SP) to address Web Single-Sign-On (SSO) issues. SAML can deal with any authentication method: it does not specify the method of authentication at the IdP, which can be anything; SAML is only responsible of the security data exchange between IdP and SP.

- SimpleSAMLphp (used by the IA2 pilot) is an open source lightweight implementation of several federation protocols written in PHP that deals with authentication, providing support for SAML 2.0. Used for deploying the IdP and the SP, configuring the federation

between the IdP and the SP, managing the exchange of identity assertions between the IdP and the SP.

- Grouper (used by the CTA, SKA and IA2 pilots) is an enterprise access management system designed for highly distributed management environments and heterogeneous information technology environments. User information can be retrieved from multiple sources (like LDAP or other databases); it is also possible to add external subjects, users whose information are not stored in any of the configured sources. Grouper V2.2.1 is used to implement the INAF CTA Attribute Authority, by SKA to implement access management, and by the IA2 pilot for managing user groups defining access rights on astronomical archive protected resources.

The IVOA pilot aims at testing the suitability of the federated cloud Infrastructure-as-a-Service (IaaS) paradigm to fulfill large-scale data and computational needs. The pilot aims at federating the EGI and CANFAR clouds, using IVOA standards to support data access and exchange. The federation model proposed is based on the assumption that the two clouds will remain independent and independently managed, but users and projects will be able to use both e-Infrastructures for data sharing and computing. Work has focused on two use cases:

- Authentication and Authorization infrastructure federation: an interoperable A&A management allows European astronomers to access CANFAR resources (data and computing) using their European (EGI Federated Cloud) credentials; at the same time, Canadian astronomers will be able to access EGI Federated Cloud resources using their Canadian credentials.

- Data federation: both CANFAR and INAF are offering virtual storage based on IVOA standards, which can be used by astronomers and data centers to store and share data exploiting an authentication and authorization mechanism based on user-password authentication and X509 credentials.

The technologies used for the implementation of the IVOA pilot include:

- Transport Layer Security (TLS) and its predecessor Secure Sockets Layer (SSL) are cryptographic protocols which allow web browsers and web servers to communicate in a secure way. This means that the data being sent is encrypted by one side, transmitted and then decrypted by the other side before processing. This is a two-way process, meaning that both the server and the browser encrypt all traffic before sending out data. SSL/TLS protocol requires Authentication. TLS is one of the

authentication mechanisms admitted by the Single Sign-On (SSO) profile specification of IVOA.

- X.509 is a standard that defines the format of public key certificates. X.509 certificates are used in many Internet protocols, including TLS/SSL, which is the basis for HTTPS, the secure protocol for browsing the WEB. An X.509 certificate contains a public key and an identity (hostname, organization, or individual), and can be signed both by a Certificate Authority and self-signed. When a certificate is signed, it can be used to rely on the public key it contains to establish secure communications with a party.

- Credential Delegation is an IVOA RESTful web service that enables users to create temporary credentials at a site and thus enable that site to perform web service actions on their behalf This feature is the key to allow services from different sites to interact. It is based on X.509 Technology.

- VOSpace is an IVOA specification based storage data service, using VOSpace-backend as a storage implementation.

- The GMS/CADC Access Control is a client and server authentication and authorization implementation for user and group management. Provides management of users and groups, and storage of group affiliation information.

For their purposes, all pilots used also other non-A&A technologies, such as Apache (versions >2.2), Tomcat (versions >6), Java (versions >7), MySQL, LDAP. The compatibility of these technologies with the A&A mechanisms used were thoroughly tested during the implementation of the pilots.

The main issues encountered were on the CentOs 6 Operating System in conjunction with Shibboleth and Tomcat 8. Several problems of incompatibilities or lost dependencies were encountered, and were solved using CentOs 7 and the technology stack proposed.

5. Python wrapper documentation and benchmark results for our optimized C/C++ mathematical Library

Python high-level programming language in the field of astrophysics and astroparticle physics is the most widely used software language, but achieving performance close to the capabilities of the hardware is a recognized challenge for researchers. In this paper, we present

performance benchmark results that we have obtained by comparing the standard Numpy Python libraries with a mathematical Python library wrapping our own highly-optimized C/C++ code. Along with performance benchmarks, we also discuss some of the best practices for Python developers.

We observed and developed numerous mathematical kernel algorithms to further improve the efficiency, during the course of ASTERICS and CTA project for developing optimized C/C++ mathematical libraries. Currently the majority of astrophysicists develop code in Python, which has triggered a need to wrap C/C++ library developed at LAPP for Python. In this section, we describe results of the performance benchmarks that we have obtained by comparing the Numpy library and optimized wrapped library developed at LAPP. We have also shown different techniques to wrap C/C++ code for python and their relative performances.

Numpy is the most widely used Python package for scientific computing. It is an open-source add-on module to Python that provides common mathematical and fast numerical routines in pre-compiled C/C++ language.

The C/C++ PLIBS_BASE open-source library is highly optimized. It contains math kernels vectorized implementations to maximize performance on each INTEL processor family. As vectorization uses SIMD registers, this library is not binary compatible between different Intel SIMD CPU instruction sets (SSE2, SSE4, AVX, AVX2). This means that the correct library needs to be installed on a system depending on its INTEL SIMD CPU instruction set. Thanks to the PLIBS_8_BASE package builder automatic code generation for the correct Intel SIMD CPU instruction set before compiling the kernels is possible. This could seem tedious at times, but it seems to be the only way to utilize the full capacity of the processor. plibs_8

plibs_8 is a Python 3 math kernel library for Intel and compatible processors. In most cases, it is faster than Numpy. It is a Python wrapper on PLIBS_8_BASE library written in C/C++. Signatures of the functions are exactly the same as that of Numpy, so developers familiar with using Numpy can easily switch to plibs_8.

Two different fashions of wrapping C/C++ library for Python have been tested. First using cython def and cpdef mixed with Python code and the second using only Python and Numpy C API.

For benchmarking, each time Python interpreter has to execute a function from a library it is needed to create a function alias otherwise CPU cycle consumption is biased. I.e:

plibs_8.tools.get_cycle () returns 380 (cycle) on system used for this benchmark. With the alias get_cycle = plibs_8.tools.get_cycle, get_cycle() returns 200 (cycle) on the same system.

Figure 4: CPU cycles consumption comparison for array sum.

Some of the best practices identified for python developers during this work are enlisted below.

- Python must be merged with Numpy or an effective wrapper on compiled code when program contains some high CPU usage computation.
- Numpy effectiveness could be improved by manually allocating array memory to be aligned on CPU SIMD register size.
- Most of Numpy performances could be outperformed by a wrapper on an optimized compiled code.
- Avoiding memory allocation drastically reduces the amount of CPU time. It is advised to reuse allocated memory as out parameter when it is possible (Numpy sum for example allows to pass an out array).
- With plibs_8 or Numpy math kernel, CPU consumption is dominated by Python function call. This can be observed from the initial values of the plots.
- Python interpreter causes more CPU consumption to get method from object/library. It is recommended to use alias in hot spots. Example: plibs_8.sum must be replaced by

an alias psum = plibs_8.sum. Using Python/Numpy API is more efficient than merging cython and Python.

- Numpy does not systematically allocate memory. It sometime uses already allocated memory for no more used object. We still need to investigated if we can use it for plibs_8array with aligned memory.
- It is recommended to execute math function on multidimensional array on the last dimension because data has to be continuous in memory. Developers have to design multidimensional array in this fashion.
- Creating and maintaining plibs_8 library that are overriding all Numpy is a significant amount of work that cannot be realized by few developers. This library will be maintained only for ctapipe project.
- Using a pitch on a multidimensional array is only slightly penalizing.
- Wrapper using only Python and Numpy C API is much faster than wrapper using cython def and cpdef mixed with Python code.
- Development of an efficient Python wrapper is not trivial

In conclusion to benchmarks development and results, we have enlisted best practices for astrophysicist and astroparticle physicists who develop algorithms in Python programming language. In addition to these best practices, it is important to use compiled libraries wrapped for Python like Numpy or plibs_8 for intensive computing. We have successfully shown that these good practices can substantially improve performances. Having said that, compiled software written in C/C++ or Fortran are still faster by a magnitude factor.

6. Use and potential improvement of the open software libraries

The activities have been focused on the use and potential improvement of the open software libraries developed by the Cosmostat group at CEA (http://www.cosmostat.org), a set of tools providing sparse methods for signal processing (already usable in SKA and other projects like Euclid).

6.1 Denoising of CTA telescope camera images

The study was performed with Monte Carlo data, first restricted to a limited number of telescopes holding the same type of camera. The camera images are made of more than 1000 pixels illuminated by a more or less uniform random background (electronics and night sky background) and a Cherenkov light signal generated from the interaction of high energy gamma-rays or cosmic rays (mainly protons) in the atmosphere. The atmospheric showers resulting from the interaction are usually seen by more than one telescope, and the

combination of several images allows estimating the gamma-ray source position in the sky from a stereoscopic reconstruction.

The difficulty of the analysis comes from:

1. The differentiation of faint gamma-ray images from the random noise on one hand;

2. The discrimination of cosmic ray from gamma ray showers, with an initial ratio of 10^5 to 1 for high intensity sources.

In CTA today, as well as in the present generation of ground Cherenkov telescope arrays, the random noise is disentangled from the signal selecting the set of camera pixels above a certain high threshold, and the close neighbours with a lower threshold above the background. In a prototype of the CTA pipeline, we have implemented this method and an image de-noising method based on the Cosmostat sparse method library (wavelet transformation). We also developed a cosmic ray/gamma ray discrimination module based on a Boosted Decision Tree, and an innovative stereoscopic reconstruction, in the objective of measuring the effect of the de-noising on the final performance (sensitivity curve).

We studied various combinations and options offered by the Cosmostat libraries to extract the gamma-ray shower signal from the random background. Preliminary results show that gamma-ray images are better identified in particular at low luminosity: the use of wavelet allows a better conservation of the light associated to the shower, improving substantially the estimation of the shower image direction in the camera. The optimisation of de-noising on protons and its capability to discriminate cosmic rays from gamma rays at the level of the camera image is ongoing.

Running the full analysis, including the discrimination between cosmic rays and gamma rays and the stereoscopic reconstruction, has shown a similar trend although a confirmation is awaited from a simulation with a much higher statistic.

6.2 Novel methods for real-time transient searches

The open software libraries developed by the Cosmostat group at CEA have been used to characterize sky images produced in real-time in CTA (or other Cherenkov ground telescopes). In particular, we have applied the aforementioned tools to time dependent signals such as gamma ray bursts (GRB) or flaring active galactic nuclei (AGN) to the on-site real time analysis. Current standard techniques monitor the count rate integrated in the entire sky image, which dilutes the signal of a potential transient in the field of view. Our innovative approach is to carry out a wavelet filtering in a data cube (x,y,time) where the signal is correlated from time slice to time slice but the noise would not. Preliminary results show that the search for transients in the wavelet filtered data cube allows us to detect much fainter transient signals than with standard techniques.

6.3 Online Analysis of High-Volume Data Streams

The full CTA array will consist of up to hundred telescopes performing simultaneous measurements. The high sensitivity of CTA will allow studying high-energy transient phenomena in the gamma-ray sky. To allow for quick follow-up studies and successful multi-wavelength campaigns it is crucial to analyse the data in real-time.

The estimated event rate for the full array will be at approximately 10 000 events per second. This provides a big challenge for the onsite computing hardware and software. To cope with the large amount of data we experimented with several technologies for real-time distributed data streaming like Apache Storm, Apache Flink, Apache Spark and Dask.

We built a complete prototype system that can read calibrated CTA raw data from Monte Carlo simulations and perform a full analysis up to high-level event lists. The prototype is build using Python and Java and techniques inspired from the CTA pipeline software. The software contains in particular modules developed at CEA that performs the camera image de-noising, shower image parametrization, the event reconstruction, the cosmic-ray background suppression and the energy estimation. This involves applying machine-learning models (like Random Forest) to the data stream in real-time.

The prototype used a single machine with the events distributed on 24 cores, and a data sample corresponding to a typical 30-minute observation.

First results show that our prototype is capable of meeting the CTA real-time requirements in parallel environments. It could also be a solution to address the problematics of high volume data streams in other ASTERICS projects.

7. Quantitative feature characterisation of CORELib

CORELib (Cosmic Ray Event Library) is a collection of simulations based on CORSIKA featuring a common set of physical parameters in order to achieve a common high statistics production.

7.1 Pilot production

A first production was launched to profile the computational and storage load. The version of CORSIKA used is 7.5000. The GRID computing infrastructure and the KM3NeT Virtual Organization have been used. The production (~1% of the total envisaged by KM3NeT) has been made in order to fill 7 energy ranges of the primary proton:

Energy range (GeV)	Number of events
200-1000	10^7
10^3-10^4	10^7
10^4-10^5	10^6
10^5-10^6	10^5
10^6-10^7	10^4
10^7-10^8	10^3
10^8-10^9	10^2

For each energy range a proper number of events has been set, in order to guarantee a proper statistical sample and to reproduce a spectrum similar to the one of cosmic rays. The selected energy spectrum has a power law with slope equal to -2. Four different sub-production have been made, changing the high-energy interaction model or the production options each time.

High energy model	Low energy model	Option	
		TAULEP	CHARM

QGSJET01	GHEISHA		X
QGSJET01	GHEISHA	X	
QGSJETII-04	GHEISHA	X	
EPOS LHC	GHEISHA	X	

A total number of 209 GRID jobs, corresponding to the same number of data files, have been executed. The events have been split in different runs in order to have output files with size less than about 1GB. The output data files sum up to a total size of about 210 GB.

7.2 Ongoing production

The ongoing production has the same characteristics of the first one except for the presence of the Cherenkov photon production. According to a dedicated preliminary evaluation, the average increase of the computation time is about 300%. The number of parallel GRID jobs has been increased accordingly to about 500 for each sub-production in order to balance the computation time.

8. Discussion of ROAst features

The ROOT analysis framework is one of the most used software for the analysis and indeed it is the “de facto” standard for high-energy physics. The goal of ROAst (ROot extension for ASTronomy) is to extend the ROOT capabilities adding packages and tools for astrophysical research.

8.1 Access to astronomical catalogues

ROAst provides a graphic support library, which rely on the ROOT graphic tools. All ROOT graphical options are automatically integrated. The general architecture relies on an intermediate abstraction layer in order to speed-up the future catalogues implementations.

The status of the current Implemented catalogues is the following:

Catalogue name	Status
UCAC4	Supported
URAT1	Under implementation
GSC-II (Guide Star Catalog)	Planned
Fermi-LAT 3FGL	Planned
TeVCat	Planned

The following table shows the supported coordinate systems:

Astronomical coordinate system	Geographical coordinate system	Time coordinate
Equatorial	N/A	N/A
Galactic	N/A	N/A
Horizontal	Lat-Long/UTM	Unix time/UTC/Local Sidereal
Equatorial rectangular (Under implementation)	N/A	N/A

ROAst features a set of coordinate and time conversion methods, which are catalogue-independent. The main methods are:

Name	Description
LL2UTM	Converts Lat-Long geographical coordinates to UTM coordinates
UTM2LL	Converts UTM geographical coordinates to Lat-Long coordinates.
UTC2UnixTime	Converts UTC to Unix time
UnixTime2UTC	Converts Unix time to UTC

Equatorial2Horizontal	Converts equatorial coordinates to horizontal ones
Horizontal2Equatorial	Converts horizontal coordinates to equatorial ones
Equatorial2Galactic	Converts equatorial coordinates to galactic ones
Galactic2Equatorial	Converts galactic coordinates to equatorial ones

Roast provides the following methods to interact with a catalogue:

Name	Description
ExtractObjectsRectange	Projects the catalogue objects on a rectangular region.
ExtractObjectsCircle	Projects the catalogue objects on a circular region
ExtractObjectsEllipse	Projects the catalogue objects on an elliptic region
WriteObjects	Writes the extracted objects into a file
ListFeatures	List of the catalogue features

Print	Shows on the standard output the extracted objects
FindObject	Seeks a/some specified object/s in the extracted ones

The extraction of the astronomical objects is performed projecting a supported catalogue on three different geometrical regions. All regions support different coordinate systems.

8.2 Skymaps (graphics)

Currently Roast implements the following plots, which can be freely customized by the user:

Astronomical coordinate system	Flat plot	Aitoff projection	Aitoff skymap
Equatorial	X	X	X
Galactic	X	Planned	Planned
Horizontal	X	Planned	Planned

8. 3 Lunar motion model

The current efforts focus on the implementation of the lunar model ELP-2000-82, in order to compute the moon position for a chosen date and time. This is almost ready at the time this text is being written.

9.Bayesian scheduler for the electromagnetic follow-up of gravitational-wave candidates

A new era for transient astronomy begins with the first gravitational-wave events detected by the LIGO detectors. One of the up-coming milestones is to connect this new type of observations with that of conventional astronomy. To this aim, an extensive electromagnetic follow-up program has been set up through collaborative agreements signed between the LIGO and Virgo collaborations and more than 80 astronomer teams around the world. Alerts are generated from the gravitational-wave data and communicated to those teams through a dedicated network.

The gravitational-wave alerts provide a number of information about the candidate event including a probability skymap that gives an estimate of the source position. For mergers of compact object binaries (of neutron stars and/or black holes) this skymap is a byproduct of the Bayesian estimation of the source parameters. It is obtained by marginalizing the posterior distribution over all parameters but the angular sky coordinates of the source. The smallest area that encloses 90 % of the posterior probability is referred to as gravitational-wave error region.

At the leading order, the size of the gravitational-wave error region is determined by the ratio between the observed gravitational wavelength by the distance between gravitational detectors in the network. The LIGO network resolve the source in a very large sky region, that extends over 1000 square degrees, typically.

Most telescopes do not have a sufficiently large field of view to cover the whole area with one field. The gravitational-wave skymap has to be covered by a mosaic often with many tiles. The exhaustive coverage of the skymap thus requires a significant amount of observing time. A number of ideas have been proposed to prioritize which tiles have to be observed first, e.g., using catalogs of near-by galaxies.

We proposed a new statistical method to optimally address this scheduling problem by coupling the gravitational-wave source model to a selection of electromagnetic counterpart models. The method identifies where and when to observe first in order to maximize the posterior probability of observing a counterpart (associated with the selected models).

The most promising counterpart models emerge from the possible association between short gamma-ray bursts and binaries with at least one neutron star. If this association is real gravitational waves could thus be coincident with the prompt and afterglow emissions of a short gamma-ray burst. Since those emissions are beamed their brightness crucially depends on the alignment of the source with respect to the observer on Earth.

Binary parameters such as distance, inclination, component masses and sky position are estimated from the gravitational wave data. Those parameters are degenerate to some degree. For this reason, the binary inclination (for instance) shows some variability across the sky. To detect a beamed counterpart, it is a good idea to observe primarily the sky regions where the binary is more likely to be face-on (i.e., inclination is zero).

We proposed the use of detectability skymaps [1] that combine all the relevant information obtained from gravitational data with counterpart model in order to define an optimal follow-up strategy. We associate a light curve model to the posterior samples obtained from the Bayesian samplers (MCMC and nested sampling) used for the gravitational-wave source parameter estimation. Using kernel density estimation, we obtain smooth sky-dependent estimates of the electromagnetic flux expected at a given time, we call detectability maps.

The detectability skymaps can then be processed by a hierarchical scheduler that **assign fields to observation slots in order to maximize the overall detection probability**.

We demonstrate that the method with a simulated binary neutron-star merger at 75 Mpc. We assume that the alert is followed in the *r*-band by the VST (limiting mag of 22.4 AB). The algorithm schedules 55 fields corresponding to 15 hours of telescope time. The counterpart is detected in the 25th field, i.e., on two days and two hours after the alert. Using the standard posterior skymap, the source position is observed five days and 22 hours after the alert, but the transient is too dim to be detected.

This algorithm is implemented as a Python module. A publication [1] that provides a proof-of-principle is about to be submitted. We now intend to implement this strategy with the LIGO/Virgo software framework.

10. Summary

This report presents a relatively diverse set of results and work in progress, but never the less some clear and very important commonalities and themes can be identified. These are:

1. Commitment and the acknowledgement of value of making the tools fit into existing frameworks, in particular Python and its package eco-system and the ROOT framework
2. Focus on computational efficiency, as in all of the facilities supported by ASTERICS there is a realisation that the scientific capabilities will be to some extent be limited by the available computing power
3. Focus on advanced statistical techniques, including Bayesian statistics, wavelet analysis and compressed representations of likelihood functions to best extract faint signals from the observed data.

4. The essential aspect of remote and distributed scientific work and data analysis and the consequent requirements for authentication and authorisation.

Although the individual observatory project requirements have driven the initial diverse set of works within this work package we expect these commonalities will provide a very fruitful possibility for collaboration in the forthcoming second half of the ASTERICS project.

11. References

[1] O. S. Salafia et al. “Where and when: optimal scheduling of the electromagnetic follow-up of gravitational-wave events based on counterpart lightcurve models”, In prep, 2017.