

A Distributed Storage for Astroparticle Physics

Alexander Kryukov*, and Minh-Duc Nguyen

M.V. Lomonosov Moscow State University, D.V. Skobeltsyn Institute of Nuclear Physics, 119992
Leninskiy gory-1, Moscow, Russia

Abstract. In this paper we present the architecture of a distributed data storage for astroparticle physics. The main advantage of the proposed architecture is the possibility to extract data on both file and event level for further processing and analysis. The storage also provides users with a special service allowing to aggregate data from different storages into a single sample. This feature permits to apply multi-messenger methods for more sophisticated investigation of the data. Users can use both Web-interface and Application Programming Interface (API) for accessing the storage.

1 Introduction

The data obtained from current and future facilities in astroparticle physics have to be stored and worked up, and potential users need access to them for analysis. The Imaging Cherenkov Telescope Arrays distributed over the globe can be mentioned here, e.g. MAGIC [1], CTA [2,3], VERITAS [4] and HESS [5]. In Russia there exists the Cherenkov timing array TAIGA [6] which also will include some imaging telescopes.

These facilities collect a tremendous volume of data. For example the annual (reduced) raw data from CTA have a volume of about 4 PB. The total volume to be managed by the CTA archive is of the order of 25 PB per year, when all data-set versions and backup replicas are considered.

Taking into account the trend to a multi-messenger approach to data processing with its potential to provide a more precise investigation of the Universe, it is very important to organize user access to the data of different setups from geographically distributed locations. Two approaches are possible to reach this goal: The first one is the multi-messenger approach (see for example [7]) and the second one is the open data approach [8,9]. Today, open access to data or, more generally, open science is becoming increasingly popular. This is due to the fact that the amount of data received often exceeds the capabilities of their processing and analysis by the collaboration. And only the involvement of all scientists interested in research in this area allows for a comprehensive analysis of the data in full scope.

Thus the development of distributed storages for astroparticle physics is very important and actual. Note that most existing collaborations have a long history and apply methods for data processing they are used to. So, our approach to the design of data storage for astroparticle physics should be based on two main principles. The first principle is that

* Corresponding author: kryukov@theory.sinp.msu.ru

there is no interference with the existing local storage. And the second principle is a treatment of user requests on a special service outside the local storage by using metadata. The communication between local storages and a user should be provided by special adaptors which define a unified interface for data exchange in the system.

The prototype of such distributed storage is developed in the framework of the German-Russian Astroparticle Data Life Cycle Initiative [10]. This initiative aim is develop of a distributed data storage system in the field of astrophysics of particles using the example of two experiments - KASCADE and TAIGA and demonstrate it works.

Below we present the architecture of the distributed data storage and briefly report the current status of the project and the nearest plans.

2 Architecture of the data storage

The main idea of the architecture of an Astroparticle Physics Distributed Data Storage (APPDS) is that there is no interference into the local storages S1...S3 (see Fig.1). Its implementation is achieved by using special adapters A1...A3. As the adapters we use the CERNVM-FS [11] file system to export the local file systems to the aggregation service (AS) in a read-only mode. This is a reliable restriction because the initial experimental data must not be modified.

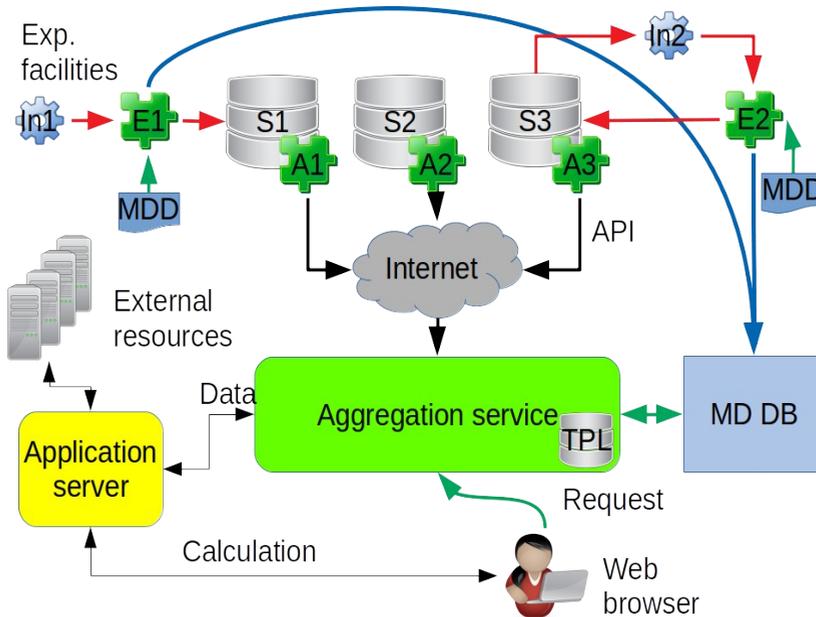


Fig. 1. The architecture of APPDS

The AS aggregates the data from the individual storages by user requests. The user requests have two types of selection criteria. The first type of the criteria is a selection on the level of all files. An example of such a condition may be the range of dates of observation of gamma sources in the sky. In this case the corresponding set of files is offered to the user. The second condition is a selection of some events from the files which satisfy the request, for example satisfying the condition of the energy range. In this case the

events are picked up from the file and aggregated to a new one. The new file is transferred to the user.

The processing of user requests is performed by the metadata database (MD DB), the main purpose of which is to specify, up to an event, in which files and where the data requested by a user are contained. Now the MD DB service is built around MariaDB [12] but we plan to migrate to TimescaleDB[13-15] after careful evaluation.

The key role in the APPDS play the extractors E1, E2. All stored data in the local storages must go through the extractors. The extractors take off the metadata (MD) from the data and store MD in MD DB. The type of extracted MD is defined by a metadata description file (MDD) which is used as input for the extractor. The MDD file is written in Kaitai Struct [16,17] with special marks pointing to elements of binary data which are MD and should be extracted.

The extractor E1 takes out the MD from the initial data. The extractor E2 extracts MD from processed data, for example after calibration of calculation of shower energy.

Thus the necessary information for treatment of user requests will be collected in MD DB service.

It is important to note that all services in the APPDS are built as microservices [18] and have well defined REST API [19]. Some services are running in Docker containers [20].

3 Status

Currently a prototype of APPDS was deployed in Lomonosov Moscow State University. The prototype consists of two local storage connected over the local network which imitate distributed storage, aggregation service and metadata service based on MariaDB. The next version of the system will also include KCDC [21] storage in KIT and storage in Irkutsk State University.

We plan to migrate to TimescaleDB after evaluating it instead of MariaDB. Most of the components of the system are written in Python.

As the first real example of using the system, the users of the CASCADE and TAIGA / TUNKA collaborations will have access to the data of these experiments, as well as to the data of the Monte Carlo simulation. However the system is developed for a broad an general use.

The work was supported by RSF No 18-41-06003.

References

1. J. Ricoa for the MAGIC Collaboration, *Nuclear and Particle Physics Proceedings*, **273–275**, 328-333 (2016)
2. *Cherenkov Telescope Array. Exploring the Universe at the Highest Energies*. URL: <https://www.cta-observatory.org/>
3. *Science with the Cherenkov Telescope Array*. Arxiv: 1709.07997
4. VERITAS. URL: <https://veritas.sao.arizona.edu/>
5. HESS, URL: <https://www.mpi-hd.mpg.de/hfm/HESS/>
6. TAIGA. URL: <https://taiga-experiment.info/>
7. A. Franckowiak, *J. Phys.: Conf. Ser.*, **888**, 012009 (2017)
8. A. Voruganti, Ch. Deil1, A. Donath, and J. King. Arxiv: 709.04217

9. S Wagner. *Gamma - Ray Astronomy in the 2020s*.
URL: https://www.eso.org/sci/meetings/2015/eso-2020/eso2015_Gamma_Ray_Wagner.pdf
10. I.Bychkov et al. *Data*, **4**, No.4, 56 (2018). DOI: 10.3390/data3040056.
11. J. Blomer, P. Buncic, G. Ganis, N. Hardi, R. Meusel, and R. Popescu. *Journal of Physics: Conf. Series*, **898**, 062031 (2017)
12. MariaDB home page. URL: <https://mariadb.org/>
13. M.J. Freedman.
URL: <https://www.percona.com/live/18/sites/default/files/slides/TimescaleDB-Percona-2018-main.pdf>
14. Michael J. Freedman. URL: <https://www.percona.com/live/18/sites/default/files/slides/TimescaleDB-Percona-2018-main.pdf>
15. Elena Štefancová, URL: <https://cds.cern.ch/record/2638621/files/evaluation-timescaledb-postgresql.pdf>
16. Kaitai Struct. URL: <http://doc.kaitai.io/>
17. I.Bychkov et al. *CEUR Workshop Proceedings*, **2267**, 563-567 (2018).
18. Microservices
19. R. Th. Fielding.
URL: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf
20. Docker home page. URL: <https://www.docker.com/>
21. KASCADE Cosmic Ray Data Centre (KCDC). URL: <https://kcdc.ikp.kit.edu/>