CEUR-WS.org/Vol-2679/short14.pdf

Data Aggregation Platform for Experiments of Astroparticle Physics[⋆]

Victoria Tokareva[0000−0001−6699−830X], Andreas Haungs[0000−0002−9638−7574], Donghwa Kang[0000−0002−5149−9767], Frank Polgart[0000−0002−9324−7146], Doris Wochele[0000−0001−6121−0632], and Jürgen Wochele[0000−0003−3854−4890][⋆⋆]

Karlsruhe Institute of Technology, Institute for Nuclear Physics,
76021 Karlsruhe, Germany
victoria.tokareva@kit.edu

Abstract. The big data revolution has overturned well-established approaches to data analysis and intensified the demand for access to heterogeneous data. Modern developed methods allow for the extraction of new knowledge from the data, which enables researchers to approach many previously unsolved mysteries. This trend, observed in many areas of human activity, is also tangible in the field of astroparticle physics. Combined analysis of various experimental data allows researchers to derive deeper insights into the processes occurring in the universe and extend the borders of our knowledge about nature.

Providing the infrastructure for such investigations is a topical issue of the astroparticle physics community.

In this report we examine a service for the aggregated retrieval of heterogeneous data from distributed storages of numerous astroparticle physics experiments. We describe its architecture, available data, principles of functioning and interaction with users and data centers.

Keywords: Astroparticle physics · Data engineering · Data curation · Asynchronous data processing · Big data.

1 Introduction

Since the moment of their discovery in 1912, cosmic rays (CR) have been studied quite thoroughly. Nevertheless, many mysteries related with them remain the subject of active community research, including CRs spectrum and

[⋆] This work was supported by the Helmholtz Society Grant HRSF-0027

[⋆⋆] The authors acknowledge the help of the colleagues from projects Astroparticle Physics Data Storage (APPDS) (esp. A. Kruykov[0000−0002−1624−6131], A. Mikhailov[0000−0003−4057−4511], M.D. Nguyen[0000−0002−5003−3623], A. Shigarov[0000−0001−5572−5349]), Tunka-Rex Virtual Observatory (TrVO) (esp. Dmitriy Kostunin[0000−0002−0487−0076]) and the SCC GridKa infrastructure at KIT.
Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

mass composition, behaviour of matter at ultra-high energies not achievable in terrestrial accelerators, mechanisms of CRs acceleration and propagation as well as their origin. Numerous experiments of astroparticle physics around the world record the particle cascades generated in the interactions of relativistic cosmic rays with the atmosphere of planet Earth.

A valuable trend in the search for insights into the above-mentioned matters, which includes development of methods for combined analysis of observations from multiple components (messengers) of cosmic radiation, is called Multi-Messenger Astroparticle Physics.

The other modalities for combined analysis are encouraging as well, as they allow an increase of the statistical data of observables and positively influence the accuracy of the analysis.

Thus, development of a united infrastructure for aggregated access to heterogeneous experimental data becomes of much importance. Such an infrastructure would integrate data from heterogeneous geographically distributed storages, be resilient, horizontally scalable, and support the entire data life cycle from creation to archiving or destruction, i.e., would satisfy the FAIR (findable-accessible-interoperable-reusable) [1] model of data curation.

In the framework of the international German-Russian Data Life Cycle initiative [2], a data aggregation system has been developed that provides access to data from diverse data centers located in Germany and Russia, which store data from astroparticle physics experiments. In this report the infrastructure, which aggregates data from the KASCADE Comic Ray Data Center (KCDC) [3], Tunka-133 [4] experiment and Tunka-Rex Virtual Observatory (TrVo) [5] is taken under consideration.

2 Data aggregation framework

2.1 Framework structure and components

The developed data aggregation system includes such elements as the setup (facility), data center of the experiment (storage), data extractor, adaptor, application programming interface by GRADLC (GRADLC API) and metadata database (MDDB). In a simplified form, the interaction of these elements is shown in Fig. 1.

Let us examine the elements from Fig. 1 and their interaction in more detail. A storage is a data center or a database of an experiment in astroparticle physics, where data of any reconstruction level registered by experimental setups (facilities) are recorded.

Infrastructures for data storage can be loaded with internal search queries of collaborators, processing new data, and with other computationally intensive tasks. Under these conditions, performing complex search queries on the storage side is undesirable, since it can lead to a decrease in the performance of the storage. To reduce the search payload, an approach was proposed to use a simplified database on the side of the aggregator, called metadata database (MDDB), which stores metadata for the data from experimental storages.

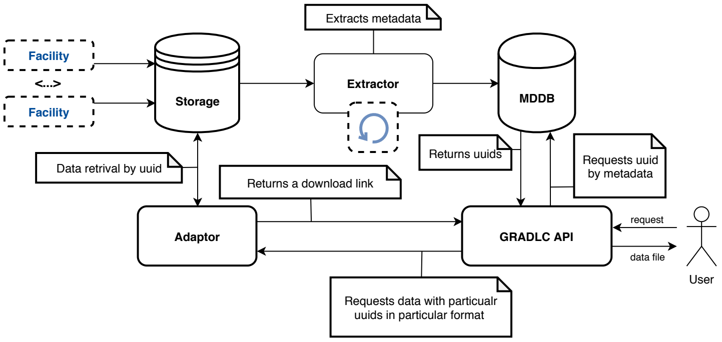


Fig. 1. The architecture design of the GRADLC platform

This approach is based on a two-level metadata system described in [2], that includes physical level metadata, such as datetime, file type etc. and event level metadata, such as event_id, setup, energy, and many more.

A program called extractor is used to retrieve information from the repositories to MDDB. The schedule of running the extractor depends on the frequency of data updates in the repository.

When a request is received to retrieve data from the system, a search for the relevant parameters is performed in MDDB in order to find universally unique identifiers (UUID) that correspond to the criteria of the events. Next, a list of unique identifiers for each experiment is passed to the aggregator, which connects to the repository, extracts the necessary data and transfers it to the aggregator.

The interaction of the user with the system and with individual components within the system is organized through the API, which will be discussed in detail in section 2.4.

It is important to note that in order to maintain simplicity, Fig. 1 shows the interaction in the case of a single data storage. However, the approach we developed allows interaction with multiple repositories, and it is easy to add new ones. In this case, a custom extractor and adapter correspond to each new data storage.

Currently, the users of the service have access to the data of Tunka-133 experiment, KCDC, and TrVo. Users of the APPDS [6] interrelated project also have certain access to the data from the TAIGA HiScore and TAIGA IACT setups [7]. The available data will be discussed in more detail in the next section.

2.2 Data available

Currently, the system provides access to the data from the following experiments: KASCADE, KASCADE-GRANDE [8], LOPES [9], Tunka-133 [4], Tunka-Rex [10], special compilations of these data (the dataset COMBINED that includes data from KASCADE and KASCADE-GRANDE experiments, based on the analysis by Sven Schoo [12] and published in the KCDC PENTARUS 1.0 release [11]), as well as simulations for some of these setups. The observables used for making basic cuts are commited into the MDDB. Thanks to a two-level data access model, users can access extremely heterogeneous data like Tunka-Rex binary traces, ROOT files with simulation results or high-level reconstruction data, e.g. for the Tunka-133 or KASCADE-GRANDE experiments. Some data, like raw data traces or energy deposits for separate detectors, are not included into available cuts.

Table 1. Possible data cuts for available experimental data

Setup	Data cuts available in GRADLC aggregator
KASCADE	Datetime, Energy, Electron number, Muon number, Zenith, Azimuth, Core distance, Shower age
GRANDE	Datetime, Zenith, Azimuth, Number of charged particles, Muon number, Core distance, Shower age
COMBINED	Datetime, Energy, Zenith, Azimuth, Core distance, Electron number, Muon number, Shower age
LOPES	LopesCompID, Datetime, Azimuth EW, Azimuth NS, CCheight EW, CCheight NS, ConeAngle EW, ConeAngle NS, EfieldMaxAbs, Eps EW, Eps NS, Elevation EW, Elevation NS, Eta EW, Eta NS, Geomagnetic Angle, Geomagnetic AngleG, NCCbeamAntennas EW, NCCbeamAntennas NS, Reconstruction, RmsCCbeam EW, RmsCCbeam NS, Xheight EW, Xheight NS
Tunka-133	Datetime, Energy, Zenith, Azimuth, Core distance, X_max
Tunka-Rex	Datetime, station_id

A system of joint data retrieval allows the user to upload data from several setups in a overlapping range of values. The list of available cuts for all the data available through the site is presented in table 1. The available range of values can be seen in the GRADLC API official documentation [13].

A list of possible selection parameters for simulations is presented in table 2.

Table 2. Possible parameters for simulations retrieval

Parameter	Posible values
Datasets	KASCADE, KASCADE-GRANDE, COMBINED
Particles	'gamma', 'proton', 'helium', 'carbon', 'silicon', 'iron'
Models	QGSjet-II-02, QGSjet-II-04, EPOS 1.99, EPOS LHC, SIBYLL 2.1, SIBYLL 2.3, SIBYLL 2.3c
Energy	1.0×10^{14} to 1.0×10^{18} eV, 5.62×10^{17} to 3.16×10^{18} eV

To analyse the presented data, one can use the analysis framework [14], integrated into KCDC also within the GRADLC project. A more detailed description of the data can be found on the official websites of the experiments and related data centers or virtual observatories, as well as in the official articles of the mentioned collaborations.

2.3 Asynchronous data processing and interaction with storages

Let us examine in more detail the behaviour of the system when it receives a data retrieval request. In Fig. 2 the user sends a request to the aggregator using the GRADLC API. On the aggregator side, a unique request identifier is generated, that is later used by the system to perform all actions associated with its processing. After that, the request is also added to the aggregator database with the request status of ”Scheduled”, and the request identifier is returned to the user to allow further actions associated with it (see section 2.4).

On the backend, the scheduler daemon notices that the new query has been added to the aggregator database, and starts its processing. For this purpose, if there are free computing resources on the server, a process is created to retrieve the requested data from the storages, described in more detail in Fig. 3. The status of the request is changed to ”Running”. It should be noted that the length of the queue and the number of simultaneously launched tasks are limited both on the side of the aggregator and on the side of the storages for technical reasons, and this may constitute a bottleneck of the system.

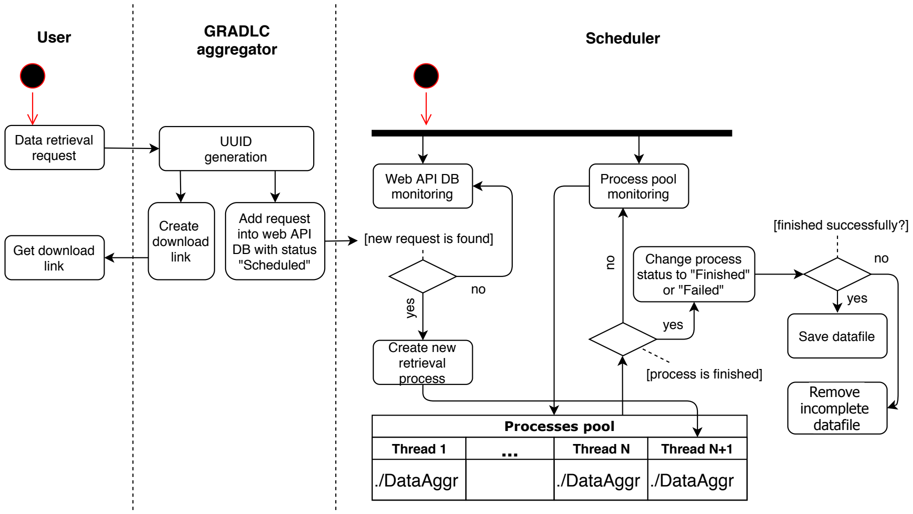


Fig. 2. Action diagram of the asynchronous parallel data processing.

The scheduler monitors running processes and upon completion of the process changes its request status in the system to ”Failed” in case of the failure or to

”Finished” in case of successful completion. For successfully completed processes, the query results are written to the server and archived for further download by the user. For the unsuccessful processes, intermediate files created during the processing are removed.

Let’s take a closer look at the process of data extraction by user request, shown in Fig. 3.

The scheduler passes the query parameters to the instance of the Data aggregator process that executes the query to MDDB and receives the UUID list corresponding to the specified parameters as its response. Next, the aggregator request the adapters of the necessary data storages for direct upload the necessary experimental data.

It should be noted that new repositories can be included quite easily: this requires a relatively easy-to-write repository adapter, a metadata extractor 2.1 and a repository data indexation with UUID.

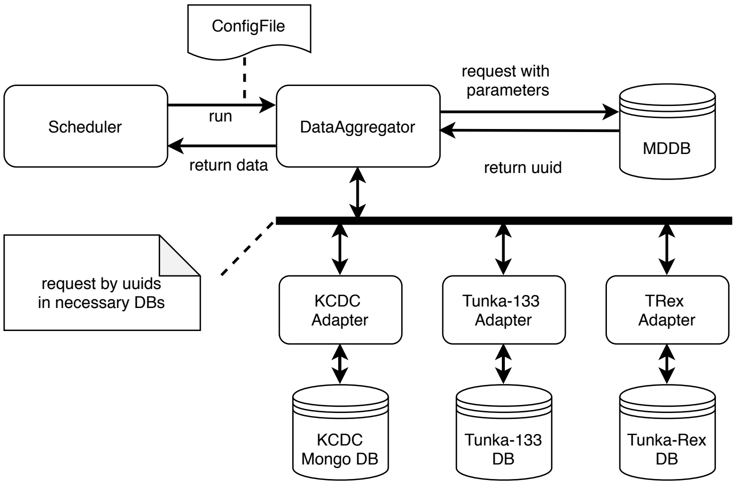


Fig. 3. Aggregation of distributed heterogeneous data from distributed data storages

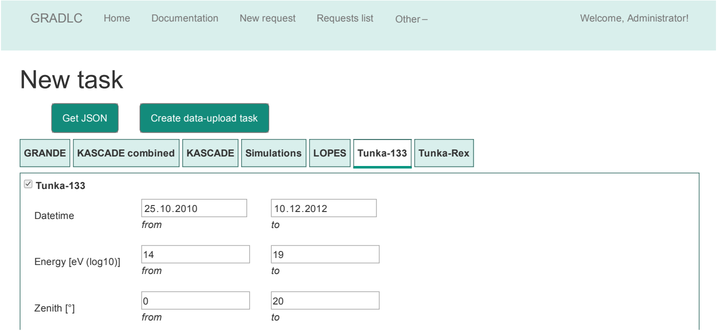
2.4 Interfaces

Application programming interface In order to provide user interaction with the aggregator an application programming interface GRADLC API was implemented, employing JSON-RPC 2.0 [15] remote procedure call protocol with data transmission over http.

There are five methods one can request through the aggregator API: data extraction, request status, list of requests, cancelling request, and data download. One can find examples of the request and detailed explanations in the official documentation [13].

Two days after the request was made related data is deleted from the server automatically and the query gets status ”Expired”. There are six possible statuses a query can have in the system: ”Scheduled”, ”Running”, ”Finished”, ”Failed”, ”Expired”, and ”Deleted”.

Web user interface For achieving a better user experience an extended web graphical user interface (Web GUI) was developed. Some examples of user interaction with it are shown in Fig. 4-5.



☑ Tunka-133

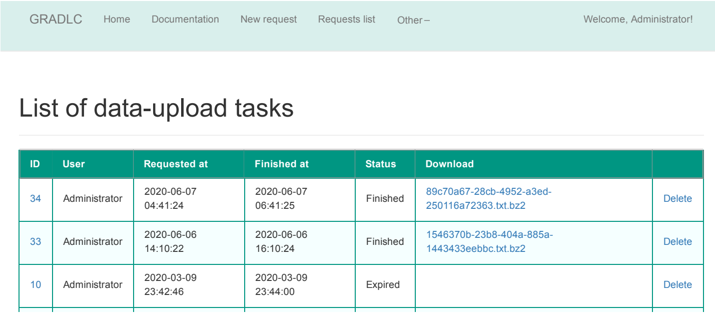
Datetime	25.10.2010 *from*	10.12.2012 *to*
Energy [eV (log10)]	14 *from*	19 *to*
Zenith [°]	0 *from*	20 *to*

Fig. 4. A new data upload query building in the GRADLC web user interface.

Fig. 4 shows an example of creating a data upload request. To do so one proceeds to the ”New task” window of the web GUI. Then, one opens the tab with the name of the experiment of interest, ticks its name inside the tab and specifies the desired values of the request parameters. Once all the desired values are chosen in all the necessary tabs, a request can be send by pressing the ”Create data upload task” button. To automate data upload, it may be useful as well to be able to generate correct JSON-RPC request from GUI. That can be done by pressing the ”Get JSON” button after the parameters selection.

To manage one’s data upload requests, the ”Requests list” section of the web GUI is used, which is shown in Fig. 5. Here one can obtain the list of all data upload queries that one has been made. In the table one can see all requests, their status, start and completion times. A request can be cancelled by clicking the ”Delete” link in the interface. Clicking on the task id shows the details about the requested data.

It is worth noting as well that the web GUI provides the user some additional possibilities, such as account management in the system (i.e. registration, login, password changes, sign out) and documentation. The web interface can be especially valuable for improving the user experience for a general audience without strong programming background, who feel more familiar interacting with graphical interfaces. Thus, the developed aggregation system can be used for outreach purposes.



List of data-upload tasks

ID	User	Requested at	Finished at	Status	Download	
34	Administrator	2020-06-07 04:41:24	2020-06-07 06:41:25	Finished	89c70a67-28cb-4952-a3ed-250116a72363.txt.bz2	Delete
33	Administrator	2020-06-06 14:10:22	2020-06-06 16:10:24	Finished	1546370b-23b8-404a-885a-1443433eebbc.txt.bz2	Delete
10	Administrator	2020-03-09 23:42:46	2020-03-09 23:44:00	Expired		Delete

Fig. 5. Request management in the GRADLC web user interface.

3 Outlook

The developed software supports aggregation of heterogeneous data from a variety of geographically spread data storages mentioned above.

The valuable features of the service, such as faster data retrieval employing a metadata data search concept, multiple filters for data search, asynchronous multithread data processing as well as different possibilities of interaction between the system and the user make it a valuable product, which could be used as a service for automatic data retrieval in large scale research projects as well as a stand-alone application for individual outreach projects.

Our future plans include employing advanced data management tools for message broking and system status monitoring, as well as possible integration of other physical experiments into the GRADLC data aggregation platform.

More information about the project can be found at the project page [16].

References

1. Haungs, A.: Towards a global analysis and data centre in Astroparticle Physics. In: this proceedings, pp. ?–?. Publisher, Location (2020)
2. Bychkov, I. et al.: Russian-German Astroparticle Data Life Cycle Initiative, Data J. **3**(4), p. 56 (2018)
3. KASCADE Cosmic Ray Data Centre, https://kcdc.ikp.kit.edu. Last accessed 27 June 2020
4. Antokhonov, B.V., Berezhnev, S.F., Besson, D. et al. TUNKA-133: A new array for the study of ultra-high energy cosmic rays. Bull. Russ. Acad. Sci. Phys. 75, 367 (2011).
5. P.A. Bezyazeekov et al. - Tunka-Rex Collaboration, Proceedings of the 3rd International Workshop on Data Life Cycle in Physics, Irkutsk, Russia, 2019, CEUR-WS 2406 3 (2019).
6. Kryukov, A.: AstroDS - A Distributed Storage for Astrophysics of Cosmic Rays. Current Status. In: this proceedings, pp. ?–?. Publisher, Location (2020)

7. Budnev, N. et al. - TAIGA Collaboration: TAIGA the Tunka Advanced Instrument for cosmic ray physics and Gamma Astronomy - present status and perspectives. JINST **9** (2014), C09021
8. Homepage of KASCADE-Grande, https://web.ikp.kit.edu/KASCADE/. Last accessed 27 June 2020
9. LOPES — A LOFAR Prototype Station Homepage, https://www.astro.ru.nl/lopes/. Last accessed 27 June 2020
10. Schröder, F. G. et al.: Tunka-Rex: Status, Plans, and Recent Results. In: EPJ Web of Conferences 135, 01003 (2017)
11. KCDC Announcements - ChangeLogs, https://kcdc.ikp.kit.edu/announcements/changeLogs/. Last accessed 27 June 2020
12. Schoo, S.: Energy Spectrum and Mass Composition of Cosmic Rays and How to Publish Air-Shower Data. PhD thesis. KIT, Karlsruhe (2016)
13. GRADLC API Documentation, http://141.52.67.147:55000/web/doc/. Last accessed 27 June 2020
14. Polgart, F.: Analysis framework for KCDC. In: this proceedings, pp. ?–?. Publisher, Location (2020)
15. JSON-RPC 2.0 Specification, https://www.jsonrpc.org/specification. Last accessed 27 June 2020
16. GRADLC project web page, http://141.52.67.147:55000. Last accessed 27 June 2020