Building a Web-Based Interface for Mapping Liquid Argon Detector Components

Mohammed VI Polytechnic University, Ben Guerir, Morocco
CERN, Geneva, Switzerland

Author: Hiba Khey

Supervisor: Huacheng Cai

August 24, 2023

Abstract

In this report, I present the outcomes of my summer internship project conducted at CERN as a member of the ATLAS liquid argon team. The project focused on developing a web-based interface for mapping information of the liquid argon detector components. As one of the two general-purpose detectors at the Large Hadron Collider (LHC), the ATLAS detector holds a pivotal role in advancing our understanding of particle physics. The precise calibration of its liquid argon system stands as a critical necessity for accurate energy measurements of electrons and photons, essential to the success of experiments conducted at the LHC. Specifically, the LAr calorimeter is tasked with both electromagnetic and hadronic calibrations, underlining its significance. Through the integration of technologies such as React, MySQL, and TypeScript, the developed interface enables users to access and visualize detector mapping data conveniently. The project involved both front-end and back-end aspects, contributing to a comprehensive understanding of web development and database management. The report outlines the objectives, implementation details, challenges faced, and outcomes achieved during the course of this internship, demonstrating the significance of the project within the context of ATLAS research.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Huancheng Cai, for his invaluable guidance, support, and mentorship throughout the course of this project. Their expertise and insights have been instrumental in shaping the direction and outcomes of this work.

I extend my heartfelt appreciation to the ATLAS liquid Argon team at CERN for their collaboration, assistance, and access to the LArId.db dataset. Their willingness to share their knowledge and resources has been essential to the success of this project.

I am also thankful to CERN for providing an environment that fosters scientific discovery and innovation. The opportunity to work within this esteemed institution has been a privilege that I deeply appreciate.

This project would not have been possible without the collective efforts of all those mentioned above, and I am truly grateful for their contributions.

Contents

1 Introduction: The ATLAS Liquid Argon Detector

The ATLAS detector is a remarkable instrument constructed for particle collider experiments. With dimensions comparable to the Eiffel Tower's weight, ATLAS measures 46 meters in length, 25 meters in diameter, and is situated 100 meters below ground level. Comprising three major systems – the inner detector, calorimeter, muon spectrometer, and forward detector – the detector records trajectory, momentum, and energy of particles produced in high-energy collisions. A magnetic system bends charged particles' paths for precise momentum measurement.

Particle beams with energies up to seven trillion electron-volts from the Large Hadron Collider (LHC) collide at ATLAS' center, generating collision debris. Despite over a billion interactions per second, only a tiny fraction are recorded for further analysis. ATLAS tracks and identifies particles to study a wide range of physics phenomena, from the Higgs boson to dark matter candidates.

The Front-End Boards (FEBs) play a crucial role in signal processing. Calibration boards simulate detector signals, while Tower Builder Boards (TBB) sum analog signals. The Liquid Argon Trigger Digitization Boards (LTDB) digitize supercell signals, facilitating further analysis. The Back-End Readout Drivers (RODs) store FEB readouts, while Digital Signal Processors (DSPs) process digitized signals.

For a virtual walk through the ATLAS detector, follow this link: Virtual Walk in the Detector

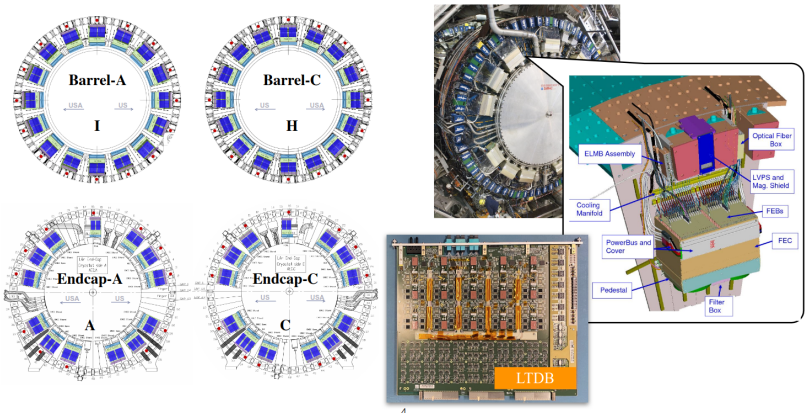

Figure 1: Front-End of the ATLAS Liquid Argon Detector

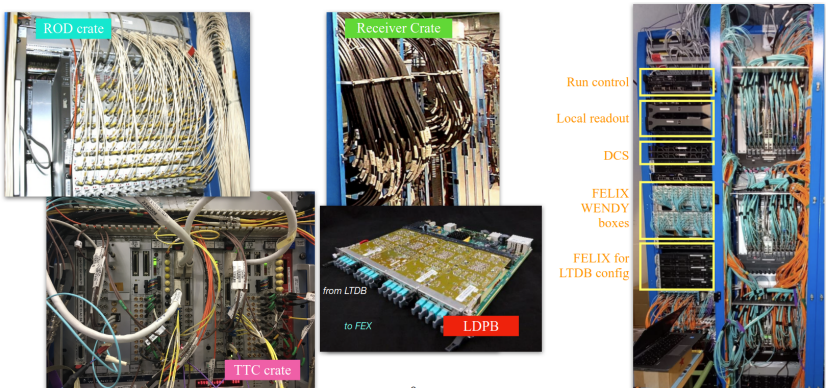

Figure 2: Back-End of the ATLAS Liquid Argon Detector

2 Motivation of LArIDTranslator

The LAr system includes 178,000 cells on the detector and another thousands of electronics corresponding for data readout, hardware connection, and triggering. Enter LArIDTranslator, a pioneering web-based mapping tool designed to streamline user access to crucial detector component information.
The ATLAS trigger system takes on the challenging task of managing this data torrent. This mechanism operates in two distinct stages: firstly, the first-level hardware trigger rapidly winnows down event options within microseconds, while the second-level software trigger meticulously dissects these events in just 200 microseconds. This process culminates in the selection of around 1000 events per second for in-depth analysis offline.
In recognition of the ever-growing importance of this endeavor, we propose an innovative step forward. We envision the creation of a JavaScript-based web application, designed to seamlessly replace the existing LArIdTranslator available at CERN's LArIdTranslator.
The new application carries with it a host of compelling features:

1. **Robust Backend Stability:** Central to the project is the establishment of a dependable backend that taps into data from dbod, a MySQL server under the diligent care of CERN IT.

2. **Empowering API Support:** This backend will wield an API (HTTP based) designed to expedite queries, providing swift and efficient access for other applications reliant on LArId.db.

3. **Cutting-edge Front-end Technology:** The front-end aspect of the application is to be meticulously crafted using TypeScript and React, technologies renowned for their steadfastness, speed, and user-centric design.

4. **Future-proof Phase-II Compatibility:** The application's significance is further underscored by its commitment to support phase-II LArId.db, ensuring its relevance as technology and needs evolve.

As we set forth on this ambitious journey, the development will unfold within the confines of the pc-emf-hls-01 environment.

3 Server Backend: MySQL Server on Demand

In the context of the proposed project, a robust server backend forms the cornerstone of the architecture. This backend is designed to seamlessly interact with data sourced from dbod, a MySQL server meticulously maintained by CERN IT.
The operational aspects of the backend server are encapsulated by the following points:

1. **Stability and Accessibility:** The backend guarantees stability, providing an unswerving link to data residing in the dbod MySQL server. This link is critical for efficient data retrieval and manipulation.

2. **Empowering Data Retrieval:** The backend is fortified with an API that employs the HTTP protocol to facilitate swift and responsive query commands. This API is tailored to swiftly address queries from other applications reliant on the LArId.db dataset.

The intricate interplay between the front-end and the backend is facilitated by the Express framework, which furnishes the necessary tools to create an interactive web application. The integration of MySQL functionalities into this architecture is made possible through the "mysql" package.
The architectural blueprint comes to life through a code snippet that provides a window into the pragmatic realization of the backend server. Within this framework, Express is employed to construct a foundation for the server, while the "mysql" package forms the bridge connecting the server to the dbod MySQL server. The API endpoints, adeptly structured, orchestrate the flow of data retrieval based on specified parameters. From managing distinct columns to handling filters and values, the backend navigates the intricacies of data access, all while ensuring responsiveness and stability.
To illustrate the interaction between the frontend and backend, consider the following API request:
`https://atlas-larmon-dev.cern.ch/dev/LArIdTranslatorBackend/data?columns=OFF_ID,ONL_ID`
The API request aims to retrieve data columns for **OFF_ID** and **ONL_ID**. In the backend, the server processes this request and generates the corresponding MySQL query:

SELECT OFF_ID, ONL_ID FROM LArId.db

The API endpoints effectively translate user requests into database queries, showcasing the intricate interplay between the frontend and backend components.
The culmination of these backend efforts ensures that the application can seamlessly retrieve and provide valuable insights from the intricate LArId.db dataset. The backend's role in accessing, processing, and delivering data is a linchpin of the proposed application's functionality, contributing significantly to its potential impact on the broader scientific community.

4 Empowering Data Accessibility and Usability: Frontend Implementation with React, TypeScript, and MUI

The frontend of the application utilizes a combination of powerful technologies – React, TypeScript, and the Material-UI framework (MUI). This triumvirate empowers the ap-

plication to deliver a responsive and user-friendly interface for accessing the LArId.db dataset.

The code exemplifies the integration of these technologies. It begins by importing essential components such as DataGridPremium for displaying data grids and various elements from MUI for building a visually appealing user interface. The interface incorporates chips, text fields, and layout components to create a coherent user experience. Within the TableContentMui component, the application's core functionality unfolds. It fetches data from the backend by constructing requests based on selected columns and filter parameters. The data is then processed and transformed to ensure compatibility with the display grid. Advanced features, such as copying cell values to the clipboard, are seamlessly integrated for enhanced user interaction.

Furthermore, the application dynamically adjusts column visibility based on the user's selections, enhancing data exploration. This adaptive behavior, combined with paginated data presentation and filtering capabilities, offers a streamlined way to navigate and extract insights from the extensive LArId.db dataset.

In the MainLayout component, the application's layout structure takes shape. It encapsulates the stacks, header, and the TableContentMui component. The stack component facilitates user-driven column selection, allowing users to tailor the displayed data to their requirements.

The collaborative efforts of React, TypeScript, and MUI culminate in an interface that not only efficiently accesses and presents the LArId.db dataset but also provides an intuitive and customizable experience for users. This frontend framework aligns with the project's overarching goal of enhancing data accessibility and usability for the scientific community.

5 Application Demo

In this section, we provide a brief demonstration of the developed JavaScript-based web application using React, TypeScript, and MUI for the frontend. The application offers seamless access to the intricate LArId.db dataset, facilitating data retrieval and analysis for researchers.

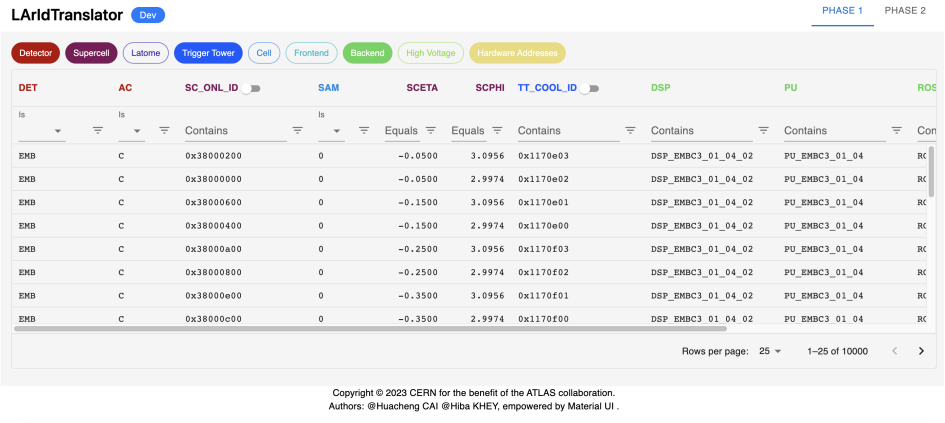

Figure 3: Screenshot of the Application

To experience the functionality firsthand, you can access the application by clicking

here.
In Figure 3, you can see a screenshot of the application interface. The interface offers a user-friendly data grid that dynamically updates based on user-selected columns and filter criteria. Users can interact with the data, copy cells to the clipboard, and efficiently navigate the dataset.
The development of this frontend application marks a significant step towards enhancing data accessibility and usability within the scientific community, catering to the complex needs of researchers working with the ATLAS experiment data.

6 Optimizing Data Retrieval for Large Datasets

Fetching and processing data from a large dataset can be a resource-intensive task, especially in the context of a scientific application like ours. As the ATLAS experiment generates immense amounts of data, optimizing the retrieval process becomes crucial to ensure responsive and efficient user experiences. In this chapter, we delve into the optimization strategies that have been employed to expedite data retrieval while dealing with the massive LArId.db dataset.

6.1 Limiting Rows for Responsiveness

One of the key challenges in handling a large dataset is managing the sheer volume of rows. To ensure optimal performance and responsive interaction, we decided to limit the number of rows displayed to the user at a time. This approach prevents overwhelming the frontend and enables quicker rendering of the data grid. For our application, we have set a cap of 10,000 rows per query, striking a balance between data comprehensiveness and performance.

6.2 Backend Query Optimization

Efficient backend queries are essential for fast data retrieval. To optimize the query execution, we have implemented several strategies:

- **Column Selection:** Instead of retrieving all available columns, we allow users to select specific columns they are interested in. This minimizes the amount of data transferred from the backend to the frontend.
- **Filtering:** Users can apply filters to narrow down the dataset before retrieval. By leveraging backend filtering capabilities, we reduce the amount of data transmitted over the network.
- **Query Execution Time:** We monitor the execution time of backend queries and implement query performance improvements whenever possible. This ensures that users experience minimal wait times while fetching data.

6.3 Asynchronous Data Loading

To enhance the user experience, we employ asynchronous data loading techniques. As the user interacts with the frontend, the application fetches data from the backend in the background. This approach allows users to continue working with the existing data while new data is fetched, ensuring uninterrupted exploration and analysis.

7 Conclusion

The *LArIdTranslator* application stands as a significant contribution to the ATLAS experiment's data analysis capabilities. By providing a user-friendly interface to query and explore the LArId.db dataset, scientists and researchers are empowered to extract valuable insights from the data efficiently. The use of modern web technologies, including React, TypeScript, and Material-UI, ensures a seamless user experience.
Through the development process, we have achieved the following key milestones:

- Successful implementation of the backend server using Express and MySQL, enabling efficient data retrieval and filtering.
- Creation of a robust frontend interface utilizing React and Material-UI components, facilitating data visualization and interaction.

8 Future Work

While the current version of *LArIdTranslator* presents a solid foundation, there are several avenues for further enhancement and exploration:

- **Performance Optimization:** Continual efforts can be directed towards optimizing the application's performance, ensuring rapid response times even as the dataset grows.
- **Advanced Filtering and Visualization:** Enhancing the frontend to support advanced filtering options and interactive visualization tools can offer users more powerful ways to explore the dataset.
- **User Authentication and Access Control:** Implementing user authentication and access control mechanisms would allow different user groups to access varying levels of data based on their roles and permissions.
- **Feedback and Collaboration:** Gathering feedback from the scientific community and collaborating with researchers to understand their specific needs can drive further refinements of the application.

In conclusion, the *LArIdTranslator* application serves as a valuable tool for the scientific community engaged in the ATLAS experiment. It not only provides a streamlined approach to accessing and interpreting complex data but also sets the stage for continued advancements in data analysis and visualization within the field of high-energy physics.