Data-centric Graphical User Interface of the ATLAS Event Index Service

Julius Hřivnáč[1,*], *Evgeny* Alexandrov[2], *Igor* Alexandrov[2], *Zbigniew* Baranowski[3], *Dario* Barberis[4], *Gancho* Dimitrov[3], *Alvaro* Fernandez Casani[5], *Elizabeth* Gallas[6], *Carlos* García Montoro[5], *Santiago* Gonzalez de la Hoz[5], *Andrei* Kazymov[2], *Mikhail* Mineev[2], *Fedor* Prokoshin[2], *Grigori* Rybkin[1], *Javier* Sanchez[5], *Jose* Salt[5], *Miguel* Villaplana Perez[7]

[1]Université Paris-Saclay, CNRS/IN2P3, IJCLab, 91405 Orsay, France

[2]Joint Institute for Nuclear Research, 6 Joliot-Curie St., Dubna, Moscow Region, 141980, Russia

[3]CERN, 1211 Geneva 23, Switzerland

[4]Physics Department of the University of Genoa and INFN Sezione di Genova, Via Dodecaneso 33, I-16146 Genova, Italy

[5]Instituto de Fisica Corpuscular (IFIC), Centro Mixto Universidad de Valencia - CSIC, Valencia, Spain

[6]Department of Physics, Oxford University, Oxford, United Kingdom

[7]Department of Physics, University of Alberta, Edmonton AB, Canada

Abstract. The Event Index service of the ATLAS experiment at the LHC keeps references to all real and simulated events. Hadoop Map files and HBase tables are used to store the Event Index data, a subset of data is also stored in the Oracle database. Several user interfaces are currently used to access and search the data, from a simple command line interface, through a programmable API, to sophisticated graphical web services. It provides a dynamic graph-like overview of all available data (and data collections). Data are shown together with their relations, like paternity or overlaps. Each data entity then gives users a set of actions available for the referenced data. Some actions are provided directly by the Event Index system, others are just interfaces to different ATLAS services. In many cases, specialized views are offered for detailed data inspection, such as histograms, Venn diagrams, etc.

This paper documents the current status of the service, its features and performance. The future system evolution to the new Event Index architecture based on the Apache Phoenix is also described as well as possible extension to a more general framework for giving a new, more intuitive access to experiment data.

1 Introduction

The ATLAS experiment [1] at the LHC accelerator at CERN uses an Event Index to catalog all its events. This Event Index has existed since 2013 and its various implementations were always based on NoSQL databases around the Apache Hadoop [2] system. A new implementation is being developed for ATLAS Run 3 (2021–2024).
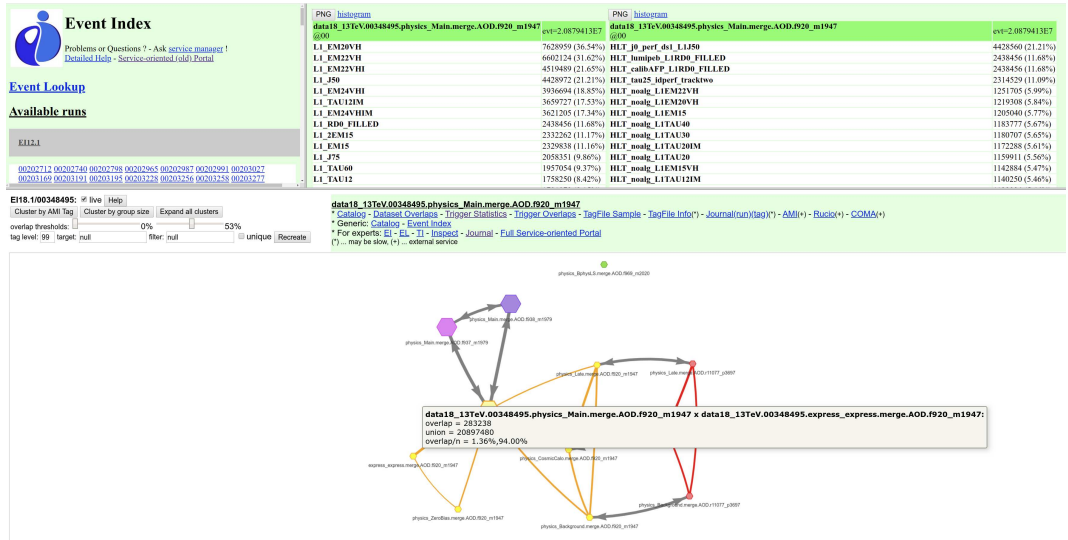
Figure 1. Data-oriented Graph Web Service.

2 Service-oriented Implementation

The original web service of the ATLAS Event Index [3] was organized by available services (like event lookup, dataset overlaps, trigger statistics, etc.). Each service gives access to all relevant data. Following user requests, the user interface has evolved into a system organized by data entity (events, datasets, collections, runs, etc.). Each entity then gives access to all available services. The underlying implementation was still organized by service type.

3 Data-oriented Implementation

A new web service implementation (under development for Run 3) [4] is organized by data entity. Two prototypes were evaluated and Prototype 2 (see 3.2) has been selected for the final implementation. A very generic dynamical web service GUI is put on top of the core application. This web service, customizable via stylesheets, is in principle re-usable for any graph-like data. The web service offers a global dynamical interactive view of all ATLAS data with relations between data entities and access to all available services. It is user-extensible. Navigation is provided via a dynamic hierarchical graph-like overview of all available data and data collections. Data are shown together with their relations, ownership, containment or overlaps. Some actions are provided directly by the Event Index system, others are interfaces to various external ATLAS services. In many cases, specialized views are offered for detailed data inspection (trigger histograms, dataset overlaps, trigger overlaps, etc.). The web service snapshot is shown in the Figure 1.
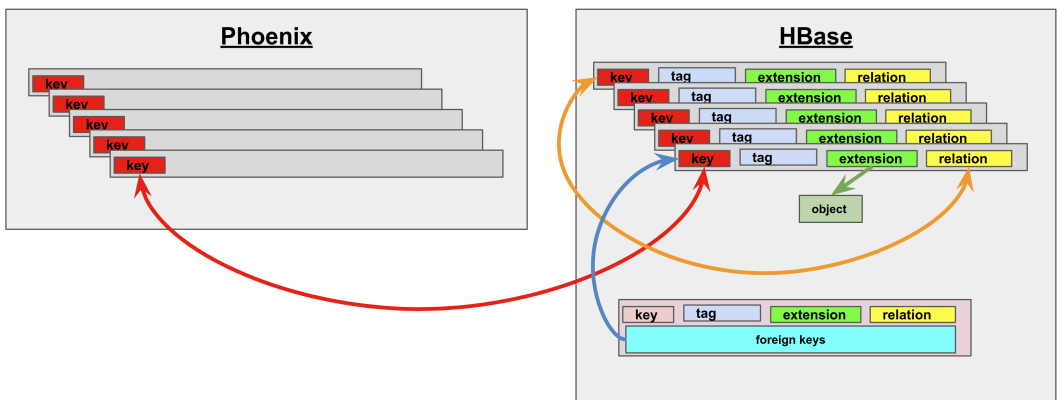
3.1 Prototype 1

Following the evolution of the Event Index Web Service towards the graph-like dynamical interface, Prototype 1 was developed to evaluate the feasibility of directly using the Graph

---

*e-mail: Julius.Hrivnac@cern.ch

Database for Event Index data. This prototype stores data directly in a JanusGraph database [5] on top of HBase [6]. Access is provided via the standard Gremlin [7] language of the TinkerPop framework [8].

By using the Graph Database directly, the prototype implementation has reproduced very performant graph-like Web Service with much less code then the previous implementation. It has been abandonned for two reasons: The relaively slow data injection and the incompatibility with other ATLAS database services mostly using SQL implementations. It has been decided to build a new prototype, bringing together SQL and Graph Database features.

3.2 Prototype 2

Figure 2. Schema of the Prototype 2 databases.

Prototype 2, which has been selected for Run 3 implementation stores data in HBase [6] tables and accesses them via the Phoenix [9] SQL API. This allows interoperability with other SQL-based ATLAS services. An additional HBase table adds more non-mandatory information. A schema of this prototype database is shown in the Figure 2.

Both databases share the same keys. User sees them both under the same interface, all data corresponding to one key is represented by one element. The additional HBase database is much smaller than the Phoenix one, as only a subset of data are included and only new information is stored there. The Phoenix database is read-only, while the additional HBase database is modifiable. The additional information can contain:

- **Simple Tags**, which can be also used in a search filter.

- **Extensions** with any object, like trigger statistics and overlap, duplicated events list, etc.

- **Relations** to other elements, which serve as a Graph Database emulation and can contain, for example, overlaps between datasets.

HBase can also contain elements without their proper Phoenix partner, called **Hubs**. They represent virtual collections of elements, like dataset tags, stream names, run numbers, project numbers, etc. They can be extended and searched in the same way as other elements. Ad-hoc virtual collections can be built using Tags.

4 Conclusion

A new prototype of the Event Index service of the ATLAS experiment at the LHC for ATLAS Run 3 is based on the combination of Apache Phoenix and a pure Apache HBase database. Its web service offers a rich, data-centric view of all ATLAS data. The whole system will become fully functional during 2020 and will replace the currently used implementation.

References

[1] ATLAS Collaboration, JINST, 3, S08003 (2008)
   DOI: 10.1088/1748-0221/3/08/S08003
[2] Apache Hadoop: https://hadoop.apache.org
[3] D. Barberis et al., J.Phys.Conf.Ser. 513 042002 (2014)
   DOI: 10.1088/1742-6596/513/4/042002
[4] Dario Barberis et al. (2020), The ATLAS EventIndex for LHC Run 3.
   Proceedings of the CHEP2019 conference J. Phys.: Conf. Ser.
[5] JanusGraph: https://janusgraph.org
[6] Apache HBase: https://hbase.apache.org
[7] Gremlin: https://tinkerpop.apache.org/gremlin.html
[8] Apache TinkerPop: http://tinkerpop.apache.org
[9] Apache Phoenix: https://phoenix.apache.org