Journal of Physics: Conference Series **664** (2015) 042045 doi:10.1088/1742-6596/664/4/042045

Integration of the EventIndex with other ATLAS systems

D. Barberis[1], S.E. Cárdenas Zárate[2], E.J. Gallas[3], F. Prokoshin[2*] on behalf of the ATLAS Collaboration

[1] Università di Genova and INFN, Genova, Italy
[2] Universidad Técnica Federico Santa María, Valparaíso, Chile
[3] University of Oxford, Oxford, UK

*Corresponding author: Fedor.Prokoshin@cern.ch

Abstract. The ATLAS EventIndex System, developed for use in LHC Run 2, is designed to index every processed event in ATLAS, replacing the TAG System used in Run 1. Its storage infrastructure, based on Hadoop open-source software framework, necessitates revamping how information in this system relates to other ATLAS systems. It will store more indexes since the fundamental mechanisms for retrieving these indexes will be better integrated into all stages of data processing, allowing more events from later stages of processing to be indexed than was possible with the previous system. Connections with other systems (conditions database, monitoring) are fundamentally critical to assess dataset completeness, identify data duplication, and check data integrity, and also enhance access to information in EventIndex by user and system interfaces. This paper gives an overview of the ATLAS systems involved, the relevant metadata, and describe the technologies we are deploying to complete these connections.

1. Introduction

Modern experiments in High Energy Physics employ complex detectors with millions of read-out channels, and gather billions of events in data taking periods that may extend to several years. These experiments are capable of producing several petabytes of data per year, with different physics conditions and data formats. It is therefore useful to have a catalogue containing information about the basic properties of event data and references to its storage location, and also providing a means of rapid search through this *metadata*.

For the LHC Run 2 taking place in 2015-2018, the ATLAS experiment [1] developed a new event catalogue that is scalable, easily maintained and modifiable if necessary: the EventIndex. It is organized in a number of modules, each of which could be developed relatively independently, making it convenient for project teams from different research institutions. The EventIndex employs the Hadoop platform for the storage and handling of event metadata, for keeping the internal catalogue and the trigger tables. The information for the EventIndex is collected both at Tier-0 and from the Grid. For data collection the EventIndex uses a producer/consumer architecture, when consumers at CERN use a messaging system for the collection of information from distributed producers. In addition, the EventIndex uses auxiliary information (trigger tables) from the condition metadata database (COMA). The Event index information is available for users through a command line client and a web interface. Figure 1 shows a sketch of the architecture and data flow in the EventIndex; more detailed descriptions of the EventIndex architecture and performance can be found in [2] and [3].

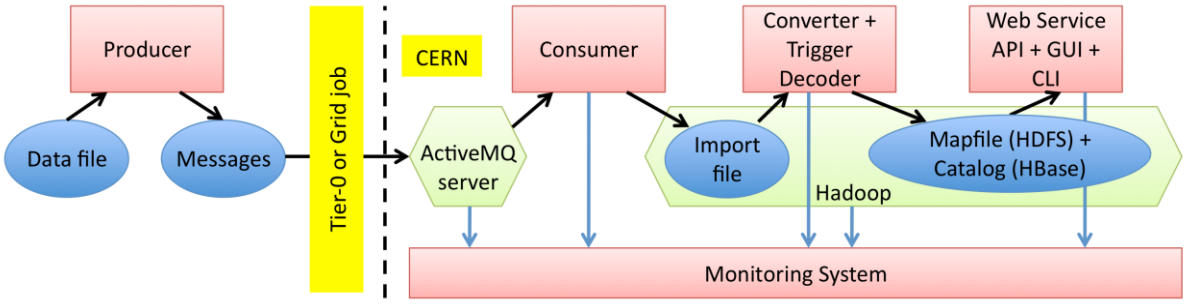

Figure 1: schematic diagram of EventIndex components and data flow.

This paper describes in Section 2 the connections between the EventIndex and the other source of metadata, COMA (Conditions Metadata in ATLAS), that are necessary to provide the users with an integrated search and information retrieval. In particular, the information about the triggers that were passed by each event needs to be decoded using the global run metadata (the Trigger Super-Master Key, or SMK) and the trigger-decoding table referenced by the SMK. All these additional metadata and the relative decoding algorithms have to be available to the EventIndex system, and updated when needed.

Section 3 outlines the monitoring tools that have been developed to have global and specialized views of the status of the servers and the processes used for EventIndex operations, from the system level, like health and load of the servers, data traffic, to the physics data level: completion of datasets and processing campaigns, overlap detection between physics streams, etc.

2. Trigger information decoding and connection to the COMA database

The data collection system gathers information from jobs running on the Grid or at Tier-0 each time new event data files, intended for permanent storage, are produced [4]. This information is sent to the core storage system that resides on a Hadoop [5] cluster at CERN. The data is stored in key-value "mapfiles," separating index keys from value data for faster access [6]. Part of the event metadata that are stored in the EventIndex is the record containing the list of online trigger conditions that were satisfied by each event.

The ATLAS trigger system for the previous years of LHC operation consisted of three sequential selection levels that reduced the event rate from the 20 MHz beam crossing rate to the 400 Hz that were acceptable to the data handling and storage systems. Level 1 (L1), dealing with the highest event rates, had to be very fast and was implemented in specialized programmable logic hardware with 256 hardcoded trigger chains. Level 2 (L2) and the Event Filter (EF) were implemented in software algorithms running on computer farms, similar to the ones used for offline reconstruction. Beginning in 2015, L2 and EF have been merged into a single High-Level Trigger (HLT) setup and the number of L1 trigger chains was increased to 512.

The trigger configuration for data taking may include 500 – 1000 trigger chains, organized in such a way as to provide the desired rate of events at the trigger output; most frequently appearing items of the trigger menu therefore have to be "prescaled" on a specific trigger level, so that only a given fraction of the events passing the prescaled chains will go on to the next trigger level or to be recorded. Some chains can even be disabled completely for some runs or portions of a run. For expert trigger studies, triggers can be run in pass-through mode, when trigger decisions are calculated but not applied. The EventIndex should have the capacity to know which events have these characteristics, not only for physics users, but also for experts who may use the EventIndex system to select these events. Prescale fractions can be changed within the same run, to follow the changes in beam setup and intensity or other data taking conditions.

ATLAS data taking evolves with time in terms of the deployed triggers which select the events recorded for offline processing. There is the "trigger menu" - a fixed set of triggers, used for each run

that takes typically a few hours of data taking, corresponding to one fill of the LHC machine. Each trigger menu contains hundreds of multi-level trigger chains, each identified by a name and assigned a unique bit. The set of such assignments is defined by a unique integer called the Super Master Key (SMK). In general, there is no guarantee that a specific chain name is assigned the same chain number from one SMK menu to another. The trigger decisions for each event are stored in the event record, and are part of the event information that the EventIndex collects. To obtain the list of trigger chains passed by a specific event, the trigger decision matrix has to be decoded and organized so that the storage in Hadoop is optimized for user queries.

The EventIndex system gets the mapping between the trigger name and the trigger bit from the COMA system [7]. COMA (Conditions Metadata for ATLAS) is based on a relational database that holds run-level metadata. It contains more than 60 tables dedicated to metadata related to data quality, trigger and prescales, data periods, loading status, general run properties, event counts, luminosity and beam related entries. The trigger data is copied and daily updated from COMA to an HBase [8] table in the Hadoop cluster. This makes the EventIndex system more self-contained, allowing it to avoid external calls from Hadoop for user access, improving its performance. The trigger menu information is stored in HBase as a table with SMK as a row key and chain names and counters as key-value pairs.

The trigger decision information for each event arrives at the EventIndex storage system as Base64 [9] compressed bitmasks, one for each trigger level. The L1 mask contains information before and after the prescale is applied (TBP, TAP), in addition to the final decision (TAV). For L2 and EF, trigger masks contain information about "physics" (PH) trigger chain selections, and also for pass-through (PT) and resurrected triggers (RS) as seen in Figure 2.

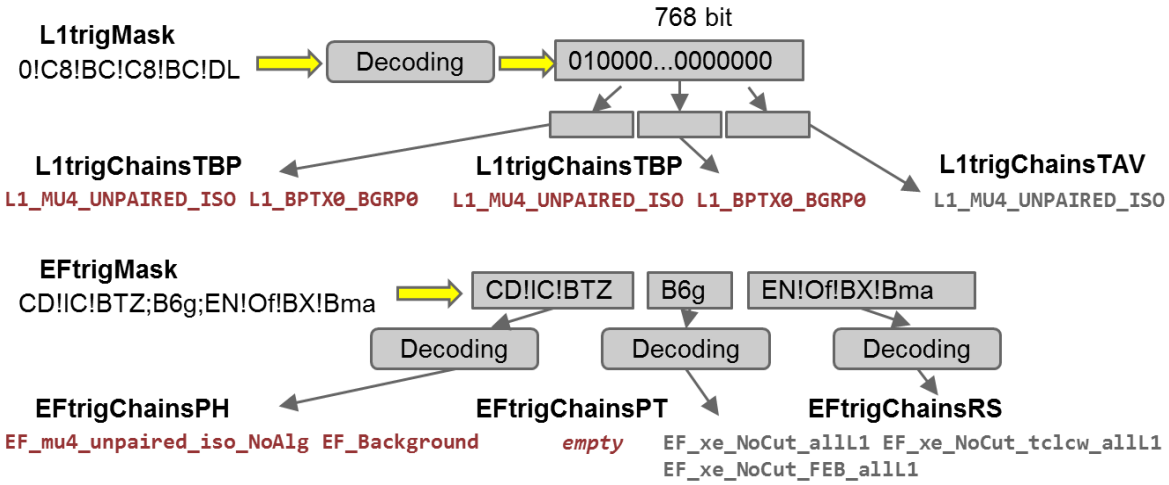

Figure 2: Trigger mask decoding

When the trigger information for a specific event has to be decoded, the trigger map corresponding to the SMK found in its EventIndex record is read from HBase into memory. This map is used for decoding of trigger information for events with the same SMK and updated if SMK has changed. The trigger mask is uncompressed and for every trigger bit the corresponding trigger chain is found using information received from HBase. Decoding results in a list of trigger chains that are passed by each event, allowing users to select events passing one or more of the chains.

The original idea was to perform trigger decoding on the fly, each time the information is requested. However, if decoding is done during the data import and the decoded trigger information is stored in the EventIndex, there is no need to perform it during every search. This simplifies code, excludes calls for HBase trigger tables during user searches and allows receiving results faster, up to 30% depending on the number of events and search type. Searches through the decoded fields can be

performed using the Hadoop machinery. The functionality of the on-the-fly decoding is nevertheless preserved.

The trigger data can be used for filtering using a specific trigger chain or a number of chains. It also possible to collect trigger statistics information on any data sample. The software used for trigger decoding is organized into a specific package in our Hadoop application software (TrigDB). It also includes tools for the verification and monitoring of the consistency of the stored information.

There is other potentially useful information in COMA, which may be used together with other EventIndex information and functionality, like data periods. A "data period" is a set of ATLAS runs taken under the same conditions. Periods are defined by ATLAS Data Preparation experts and stored in the COMA system. Periods are used in many stages of ATLAS data processing, assessment, and selection. User event samples for analysis are generally based on data periods. The EventIndex can get data period information from COMA to provide the users with period-wise selection.

Other metadata from COMA may be incorporated into the EventIndex to facilitate user searches, such as the Good Run Lists (GRL), which select runs or portions of runs during data periods that satisfy given sets of data quality criteria.

3. System monitoring

For the previous years of LHC operation, ATLAS used extensively the Service Level Status system (SLS) [10], a web-based tool that dynamically shows the basic availability information and/or statistics about CERN IT services, taking into account the dependences between them. For Run 2 a new system is in the process of being deployed [11], with ElasticSearch/Kibana [12] used for easy data access and visualization. EventIndex monitoring currently feeds both of these monitoring frameworks with system status information.

A schematic view of the EventIndex system monitoring is presented in Figure 3.

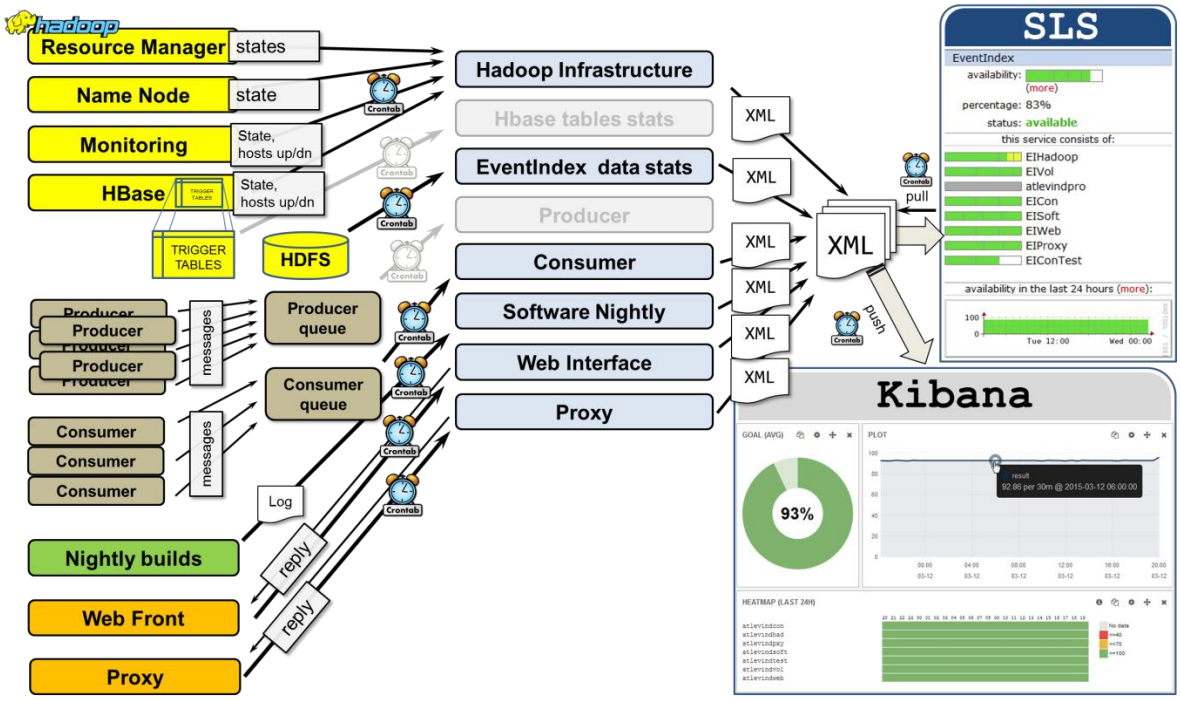

Figure 3: EventIndex monitoring.

Monitoring information is collected by *acron* jobs running on one of the EventIndex computer nodes. The components of the EventIndex system are very different and therefore are monitored in different ways:

- Hadoop components are monitored by *curl* queries that return component status and important parameters like hosts status and load.
- The volume of information stored in the EventIndex is monitored by querying the Hadoop file system.
- Trigger related monitoring is done by our TrigDB tool package, collecting tables with statistics and verifying data consistency.
- Data collection systems are monitored by collecting messages from producers and consumers. For that purpose a special monitoring queue was created. Special consumer statistics messages are generated every 60 seconds with information on data received during this period. Problems in the messaging system can be detected by comparing rates between the producer and consumer messages.
- EventIndex software is rebuilt every night; the log of this nightly build is scanned for errors by a monitoring script.
- Finally, the monitoring system checks if the EventIndex web service front-end and proxy are responding.

The gathered information is organized into *xml* files with structures that both SLS and Kibana systems can understand. The resulting files are distributed via the SLS pulling mechanism and pushing to the new Kibana monitoring server. Both systems check for updates of the *xml* files every 10 minutes. If no update is found for some component, it will be marked as unavailable and an alarm can be generated.

4. Conclusions and outlook

Development of the EventIndex started in 2013 and at the beginning of 2015 the EventIndex project is in the deployment phase, with all its components operative. The data from the previous years of LHC operation is imported, and import and decoding of the new data was started as soon as the data and condition information were available.

Trigger decoding services have been developed, tested and are currently used in data import. In addition, tools for the verification and monitoring of the consistency of the information have been developed. Extra functionality is under development, including the use of the data period information imported from COMA, and support for Good Run Lists.

The monitoring tools of the system are in operation. Data are supplied for both the SLS and Kibana frameworks, intended to be used eventually by shifters and the operation team when the system reaches complete and stable production operation.

References

[1] The ATLAS Collaboration 2008 The ATLAS Experiment at the CERN Large Hadron Collider, *JINST 3 S08003*

[2] Barberis D *et al.* 2014 The ATLAS EventIndex: an event catalogue for experiments collecting large amounts of data, *J.Phys.Conf.Ser.* **513** 042002 doi:10.1088/1742-6596/513/4/042002

[3] Barberis D *et al.* 2015 The ATLAS EventIndex: architecture, design choices, deployment and first operation experience, *J. Phys. Conf. Ser.*, these proceedings.

[4] Sanchez J *et al.* 2015 Distributed Data Collection for the ATLAS EventIndex, *J. Phys. Conf. Ser.*, these proceedings.

[5] Hadoop: http://hadoop.apache.org

[6] Hřivnáč J *et al.* 2015 QuerySpaces on Hadoop for the ATLAS EventIndex, *J. Phys. Conf. Ser.*, these proceedings.

[7] E J Gallas *et al.* on behalf of the ATLAS Collaboration 2012 Conditions and configuration metadata for the ATLAS experiment, *J. Phys. Conf. Ser.,* **396** 052033 doi:10.1088/1742-6596/396/5/052033

[8] HBase: http://hbase.apache.org
[9] Base64: http://en.wikipedia.org/wiki/Base64
[10] S. Lopienski 2008. Service Level Status - a new real-time status display for IT services, *J. Phys.: Conf. Ser.* **119** 052025 doi:10.1088/1742-6596/119/5/052025
[11] CERN IT service monitoring: https://cern.service-now.com/service-portal/sls.do
[12] Kibana: https://www.elastic.co/products/kibana