TRIGGER INFORMATION DATA FLOW FOR THE ATLAS EVENTINDEX

Mikhail Mineev [1, a], Fedor Prokoshin [2, b], Alexander Yakovlev [1]
(on behalf of the ATLAS Collaboration)

[1] *Joint Institute for Nuclear Research, Dubna, Russia*

[2] *Centro Científico Tecnológico de Valparaíso-CCTVal,*
Universidad Técnica Federico Santa María, Valparaiso, Chile

E-mail: [a] mineev@jinr.ru, [b] fedor.prokoshin@cern.ch

Trigger information is an important part of the ATLAS event data. In the EventIndex trigger information is collected for various use cases, including event selection and overlap counting. Decoding trigger information from the event records, stored as a bit mask, requires additional input from the condition database and the Monte Carlo trigger database, as trigger configurations evolve with time. Trigger information decoding depends on the run number for the recorded data and from the settings for Monte Carlo simulation. This article describes trigger information handling in the EventIndex and the interfaces used to access it.

Keywords: trigger information, ATLAS detector, EventIndex, HBase, EITrigDB library

1. Introduction

The ATLAS experiment [1] at the Large Hadron Collider (LHC) records large amounts of experimental data, analysis of which requires equally large amounts of Monte Carlo simulated data. The EventIndex [2] is a global catalogue of all ATLAS events in all processing stages. ATLAS events are stored in files grouped into datasets. They are identified by GUIDs (Globally Unique IDentifiers). Event searching in the EventIndex can be done by various parameters:

- event identifiers (run/event numbers, trigger stream, luminosity block),
- trigger information (level 1 trigger (L1) and high level trigger (HLT)),
- references (pointers) to the events at each processing step in all permanent files on storage.

The Trigger Database library (EITrigDB) is used for trigger information importing, decoding and access in the EventIndex. It retrieves the data from different trigger databases and stores in the HBase [3]. Apache HBase is an open-source, distributed, versioned, non-relational database. HBase provides Bigtable-like capabilities on top of Hadoop and HDFS. This library is written in Java. The EITrigDB core class is used by the ATLAS EventIndex core package (Atlas-Event-Index-Core) [4]. This article describes the trigger information and its handling with EITrigDB library.

2. The trigger information dataflow

The ATLAS trigger system had three selection levels during Run 1: Level-1 (implemented in specialized hardware), Level-2 (software-based analysis of information within the “regions of interest” (RoI) marked by Level-1) and the Event Filter (software-based analysis of the full event). For Run 2 the software-based triggers have been merged into a single unit: the High-Level Trigger (see Figure 1).

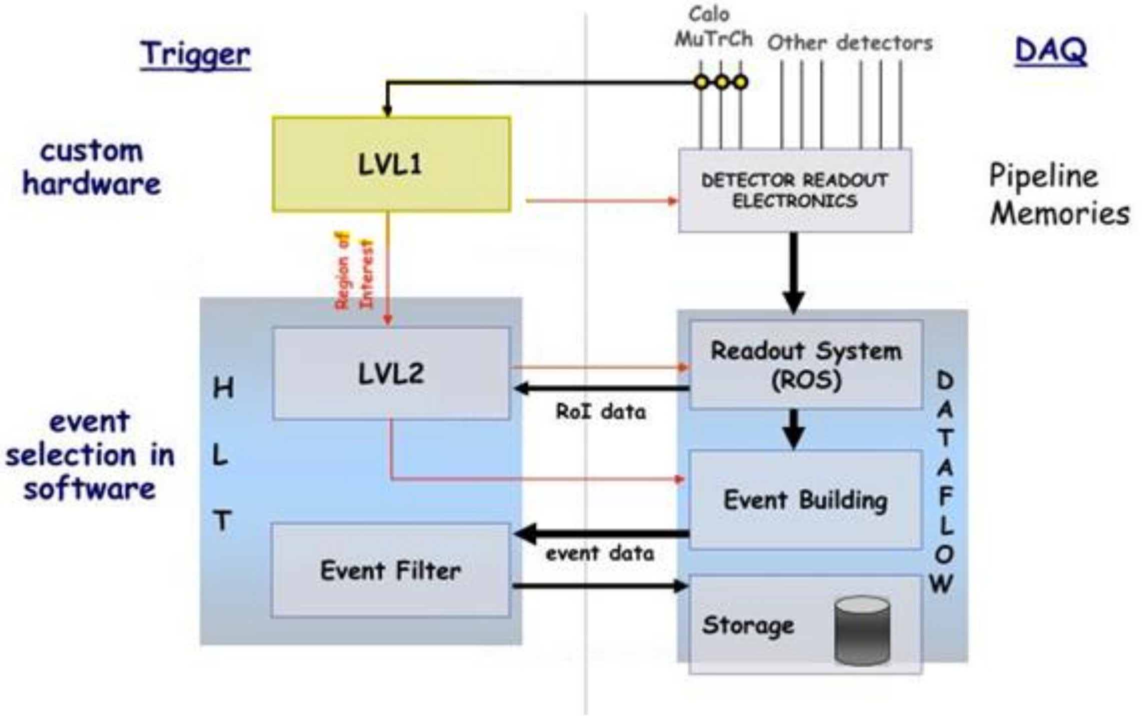

Figure 1. The Trigger-DAQ System (Thilo Pauli, CERN)

Trigger decisions are stored in data files as trigger masks encoded using the B64 algorithm. If a bit of the mask is set, this means that one of the sets of trigger conditions (trigger chains) has been passed in this event. The position of this bit in mask (chain counter) indicates which chain has been passed. Depending on the trigger configuration, the same chain counter may correspond to a different chain name. The relation is uniquely defined by the trigger table corresponding to the trigger key (or Super Master Key - SMK). Trigger masks from event record are decoded, and chain counters are converted to chain names. The list of trigger chain names obtained after decoding are then stored in updated event records. The COMA (COnditions MetadatA) [5] database in Oracle contains all trigger information for recorded data. The EventIndex uses trigger tables replicated from COMA to HBase by EITrigDB library for the trigger decision decoding. The trigger tables use SMK as key and ChainName:ChainCounter pairs as columns. If the SMK is absent in the event record it is possible to obtain it from the run number that is a part of the event ID. For this purpose Run:SMK table is replicated in the HBase from COMA. The trigger decoding data flow is presented in Figure 2.

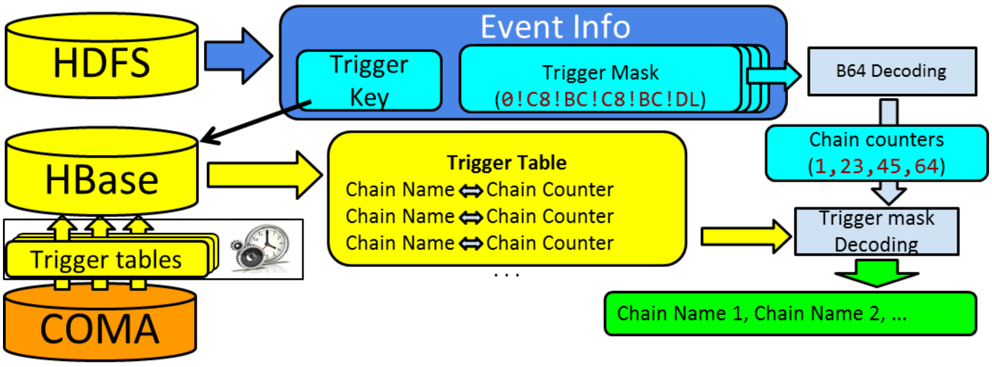

Figure 2. Trigger information decoding

The dataset trigger overlaps show how many events satisfy the different trigger conditions (more than one trigger chains are passed). The trigger overlaps on the Hadoop Data Browser Web page [6] is an example how the trigger information is used for the trigger menu efficiency investigation. The information can be presented as a table or in form of a graph.

3. The EITrigDB library update

The previous version of the EITrigDB had no simulated Monte Carlo (MC) data support. Our task was to make the EITrigDB work with the MC data. The main goals in the update were:

- add trigger information support for the MC data,
- keep the library structure the same as in the previous version.

The COMA database does not contain trigger information for the MC data. This information is stored in the Monte Carlo Trigger DB (TRIGGERDBMC). ATLAS Offline Release [7] contains a package TrigConfigSvc reading the trigger chain table. It has two versions, in C++ and in Python. The MC Trigger DB parameters (SQL query and connection parameters) are used in the EITrigDB Java code.

As in the case of recorded data, sometimes the SMK can be absent in the event record. But in this case it is not possible to use the run number (this parameter is called MC channel number in the MC datasets) because the SMK does not depend on this parameter. The trigger information appears in the MC data at the reconstruction level. That is why the SMK number can be restored using the reconstruction tag (r-tag). The r-tags are a part of the event ID, like a run number.

ATLAS Metadata Interface (AMI) [7] contains the r-tags and the information about SMK. AMI has a Web interface which can be used to browse the r-tags [8]. The Python AMI client is used to retrieve the table r-tag – SMK from AMI. This client is a part of the pyAMI package. EITrigDB calls the Python methods from Java. The r-tags corresponding to the reconstruction production step (production step = "recon") contain SMKs (see Table 1).

Table 1: R-tags and SMK

r-tag	productionStep	SMK	SMK stored in HBase
r10517	recon	2203	2203
r10210	merge	absent	-1

All SMKs are stored in the HBase. If a r-tag has no SMK than the DB returns -1 instead of positive-only SMK.

New classes were added in the EITrigDB library for MC data. They have suffix MC:

EITrigDB -> EITrigDBMC (storing and retrieving the data in/from HBase)

EITrigDBig -> EITrigDBigMC (obtaining the data from MC TriggerDB, AMI)

The added classes inherit large part of methods form the classes for the recorded data. To save the same library structure, the methods which read the data from the databases are rewritten retaining the same names. R-tags are stored in the HBase without prefix “r” (the number only). This makes it possible to work with them like with run numbers in case of the recorded data. Some new methods were added in the EITrigDBigMC class. They retrieve the SMK for a given event ID from AMI and from HBase.

4. Conclusion

The new realization of the EITrigDB library has the simulated Monte Carlo data support. It has practically the same API as in the first version. As a next step it is possible to use Java AMI client instead of the external pyAMI package.

5. Acknowledgements

We would like to thank our colleagues: Joerg Stelzer and William Panduro Vazquez for their assistance and help in this work.

References

[1] ATLAS Collaboration, The ATLAS Experiment at the CERN Large Hadron Collider, JINST 3 S08003 doi:10.1088/1748-0221/3/08/S08003, 2008.

[2] Barberis D et al, The ATLAS EventIndex: architecture, design choices, deployment and first operation experience, J. Phys.: Conf. Ser. 664 042003, doi:10.1088/1742-235 6596/664/4/042003, 2015.

[3] HBase home page: https://hbase.apache.org

[4] D.Barberis et al., The ATLAS Event Index: The Architecture of the Core Engine, 18th International Workshop on Advanced Computing and Analysis Techniques in Physics Research, Seattle, WA, USA, 21 - 25 Aug 2017, https://cds.cern.ch/record/2291669/

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

[6] F. Prokoshin et al., ATLAS EventIndex a catalog of physics data // EventIndex Workshop, 4-5 June 2018, Milan, p. 13-20, https://indico.cern.ch/event/708832/sessions/277733/

[7] Albrand S. et al., The ATLAS metadata interface, J. Phys. Conf. Ser. 219 042030, doi:10.1088/1742-6596/219/4/042030, 2010.

[8] J Fulachier et al, ATLAS Metadata Interface (AMI), a generic metadata framework, 2017, J. Phys.: Conf. Ser. 898 062001, http://inspirehep.net/record/1638497