Journal of Physics: Conference Series **119** (2008) 042031 doi:10.1088/1742-6596/119/4/042031

ATLAS Conditions Database Experience with the LCG COOL Conditions Database Project

Monica Verducci on behalf of the ATLAS Collaboration

CERN 1211 Meyrin, Geneve, Switzerland and CNAF via Viale Berti Pichat 6/2, Bologna, Italy
monica.verducci@cern.ch

Abstract. The size and complexity of LHC experiments raise unprecedented challenges not only in terms of detector design, construction and operation, but also in terms of software models and data access and storage. The nominal interaction rate of about 1 GHz at the design luminosity of 10^{34} cm^{-2} s^{-1} must be reduced online by about seven orders of magnitude to an event rate of O(100) Hz going to mass storage, and consisting of several different streams, one of them entirely dedicated to the calibration. One of the most challenging tasks will be the storage of non-event data produced by calibration and alignment stream processes into the Conditions Database at the Tier0 (located at CERN). In this work, the ATLAS Calibration Streams and the Conditions Database will be described.

1. Introduction to the ATLAS experiment at the LHC

The accelerator LHC (Large Hadron Collider) is a proton-proton collider that will run at 14 TeV in the center of the energy mass.

Along the 27 Km ring, four different detectors are placed: ATLAS (A Toroidal LHC ApparatuS), CMS (Compact Muon Solenoid), ALICE (A Large Ion Collider Experiment) and LHCb (Large Hadron Collider bphysics).

The protons, coming in roughly cylindrical bunches of few centimeters long and few microns in radius and separated in time by 25 ns, will collide in a 110 m long region, without any magnetic field. At the designed high luminosity (10^{34} cm^{-2} s^{-1}), the two proton beams will be made of 2835 bunches, with a total number of produced events per unit time (Rate) of about 1 GHz, according to the proton-proton inelastic cross section of σ = 70 mb.

The major problem of data management, that the LHC experiments, and in particular ATLAS, are going to face, is the huge quantity of data each year of about few PBytes, that must be stored and made available to all physicists of the collaboration, distributed world-wide. The distributed architecture based on Grid infrastructure has been chosen to resolve problems related to data storage capacity and data transfer among computing centers, spread in different countries, to assure data access only to the authorized users and to ensure remote resources are used effectively.

2. Event Selection (Trigger) and Streams

The major challenge faced at ATLAS is to reduce the interaction rate of about 1 GHz at the design luminosity of 10^{34} cm^{-2} s^{-1} online by about seven orders of magnitude to an event rate of O(100) Hz going to mass storage.

Divided in different levels of increased latency and complexity, the trigger chain operates a selection in the event, according to the pre-defined trigger menu. The final output, after the Event Filter (EF), consists of several streams with a well defined scope; see for details [1].

The output data from the EF requires an average 320 MB/s bandwidth connecting it to the first-pass processing facility.

Four types of streams come from the Event Filter: a ***Primary Stream***, a ***Calibration and Alignment Stream,*** *an* ***Express-line Stream*** *and a* ***Diagnostic Stream.***

The baseline model assumes a single ***Primary Stream*** containing all physics events, divided into five sub-streams, flowing from the Event Filter to Tier-0.

A ***Calibration and Alignment Stream,*** dedicated to the calibration and about 10% of the entire EF output, containing calibration trigger events (which would most likely include certain physics event classes). This stream is required to produce calibrations of sufficient quality to allow a useful first-pass processing of the main stream with minimum latency. A working target (which remains to be shown to be achievable) is to process 50% of the data within 8 hours and 90% within 24 hours and all the data within 48 hours.

An ***Express-line Stream,*** dedicated to a rapid processing, containing about 10% of the full data rate and processing as soon as possible and anyway within 8 hours. This stream has been also included to do calibration and alignment studies as well as the calibration stream; moreover, it will allow the tuning of physics and detector algorithms and also a rapid alert on some high-profile physics triggers. It is to be stressed that any physics based on this stream must be validated with the "standard" versions of the events in the primary physics stream. However, such a hot-line should lead to improved reconstruction. It is intended to make much of the early raw-data access in the model point to this and the calibration streams. The fractional rate of the express stream will vary with time, and will be discussed in the context of the commissioning.

And a ***Diagnostic Stream*** dedicated to the events causing problems at EF level. These may pass the standard Tier-0 processing, but if not they will attract the attention of the development team. They will be strongly rate-limited.

2.1. Calibration and Alignment Process

Calibration and alignment processing refers to the processes that generate "non-event" data that are needed for the reconstruction of the event data, including processing in the trigger/event filter system, prompt reconstruction and subsequent later reconstruction passes.

These "non-event" data (i.e. calibration or alignment entries) are generally produced by processing some raw data from one or more sub-detectors, rather than full raw data. The input raw data can be in the event stream (either normal physics events or special calibration triggers) or can be processed directly in the sub-detector read-out systems in special calibration runs. The output calibration and alignment data will be stored in the conditions database, and may be fed back to the online system for use in subsequent data taking, as well as being used for later reconstruction passes, [2].

Various types of calibration and alignment processing can be distinguished:

- Processing directly in the sub-detector read-out system (the RODs). In this case, the processing is done using partial event fragments from one sub-detector only, and these raw data fragments do not need to be passed up through the standard Data Acquisition chain into the event stream (except for debugging). This mode of operation can be used in dedicated stand-alone calibration runs, or using special triggers during normal physics data-taking.

- Processing in the EF system, with algorithms either using dedicated calibration triggers (identified in the level 1 trigger or in the High Level Trigger HLT). In particular, an algorithm running at the end of a chain of event filter algorithms would have access to all the reconstructed information (e.g. tracks) produced during event filter processing, which may be an ideal point to perform some types of calibration or monitoring tasks. If the calibration events are identified at level 1 or 2, the event filter architecture allows such events to be sent to dedicated sub-farms, or even for remote processing at outside institutes.
- Processing after the event filter, but before prompt reconstruction. Event byte-stream RAW data files will be copied from the event filter to the Tier-0 input buffer disk, and could then be processed by dedicated calibration tasks running in advance of prompt reconstruction. This could be done using part of the Tier-0 resources, or event files could also be sent to remote institutes for processing, the calibration results being sent back for use in later prompt reconstruction, provided the latency and network reliability issues can be kept under control.
- Processing offline after prompt reconstruction. This would most likely run on outside Tier-1 or Tier-2 centers associated with the sub-detector calibration communities, leaving CERN computing resources free to concentrate on other tasks; the detailed calibration plans for each sub-detector are still evolving.

3. Conditions and Configuration Databases

Many types of non-event data will be used during the ATLAS data taking, reconstruction and subsequent processing. These data have many different origins, and are stored in many different types of databases, see Figure1.

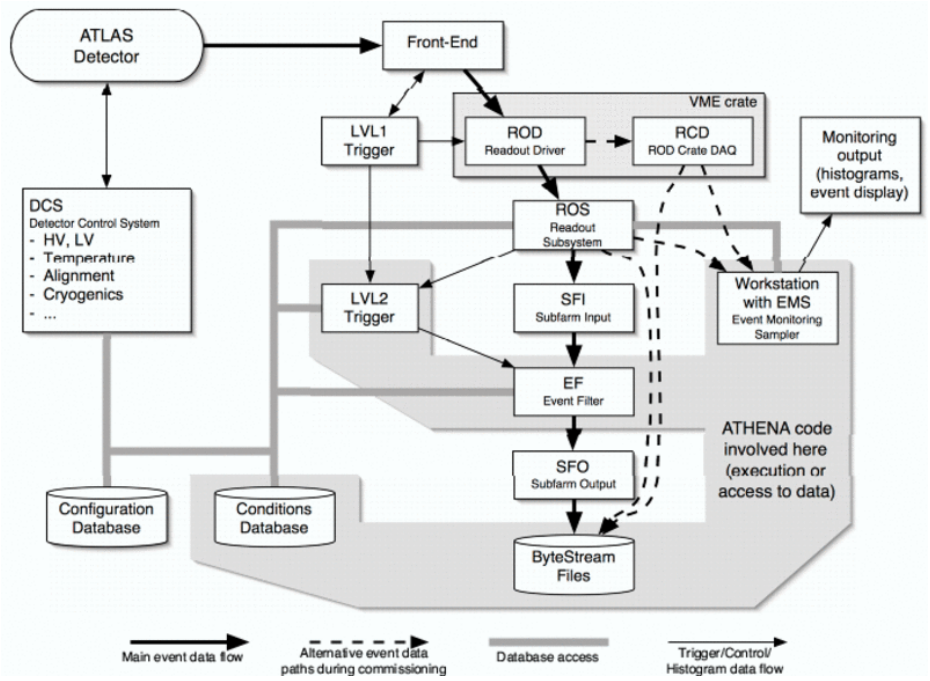

Figure 1: Data Flow of the Events and Conditions Data in the ATLAS experiment. The Configuration and the Conditions Databases are shown (on the left of the picture) and their links with the Detector Hardware and Software of the experiment. The Configuration DB is directly connected to the ATLAS Detector Control System (DCS), while the Conditions DB is wrapped in the ATHENA Framework, connected both to the Configurations DB and to the Trigger System, from where it takes all the information required by the reconstruction (ATHENA).

In the ATLAS experiment there are two database systems to store these non-event data: a Configuration Database and a Condition Database, see [2].

The Configuration Database will store all the data needed at the start of the run, including sub-detector hardware and software configuration. The Conditions Database will store all the parameters describing run conditions and logging, all the data which will be accessed offline, i.e. by the reconstruction or analysis software.

The conditions database is closely related to Configuration Database, needed to set up and run the detector hardware and associated online and event selection software.

Conditions data varies with time, and is usually characterized by an "interval of validity" (IoV). It includes data archived from the ATLAS detector control system (DCS), online book-keeping data, online and offline calibration and alignment data, and monitoring data characterizing the performance of the detector and software during any particular period of time.

3.1. Conditions Database

The ATLAS Condition Database is based on Oracle DB, all the Condition Database, in particular for the offline reconstruction, is implemented using COOL technology. COOL, an LCG product, is a library to manage conditions data in terms of Interval of Validity (IoV), versions and tags, using CORAL as backend. CORAL allows database applications to be written independently of the underlying database technology (this means that COOL databases can be stored in Oracle, SQLite or MySQL), see for more details [3].

Moreover, the COOL API has been integrated into the ATLAS online software. Several special-purpose higher level interfaces are also being developed, including the Condition Database Interface (CDI) for archiving information system (IS) data to COOL, the PVSS to COOL interface for archiving Detector Control System (DCS) data, and specialized interfaces for saving monitoring data.

The objects stored or referenced in COOL have an associated start and end time between which they are valid (IoV).

COOL data is stored in folders, which are themselves arranged in a hierarchical structure of folder sets. Within each folder, several objects of the same type are stored, each with its interval of validity range. These times are specified either as run/event, or as absolute timestamps, and the choice between formats is made according to meta-data associated with each folder. The objects in COOL folders can be optionally identified by a channel number (or channel ID) within the folder. Each channel has its own intervals of validity, but all channels can be dealt with together in bulk updates or retrievals.

COOL implements each folder as a relational database table, with each stored object corresponding to a row in the table. COOL creates columns for the start and end times of each object, and optionally the channel ID and tag if used. Several other columns are also created (e.g. insertion time and object ID), to be used internally by the COOL system, but these are generally of no concern to the user.

The payload columns (where the data are stored) are defined by the user when the table is created. In ATLAS, the payload data can be stored in the three following ways.

The payload data can be stored directly in one or more payload columns (inline data), where the columns directly represent the data being stored (e.g. a mixture of float and integer values in the columns representing status and parameter information).

In second way, the payload data (in this case a single column) can be used to reference data stored elsewhere. This reference can be a foreign key to another database table, or a reference to something outside of COOL - e.g. a POOL object reference allowing an external object to be associated to intervals of validity.

A third approach involves storing the data as an inline CLOB in the database, i.e. defining the payload to be a large character object (CLOB) which has an internal structure invisible to the COOL database. COOL is then responsible only for storing and retrieving the CLOB, and its interpretation is up to the client code.

The retrieving and storing of the data inside a reconstruction job in the Athena framework (offline reconstruction framework) is possible using the IOVService, a software interface between the COOL DB and the reconstruction algorithms via IOV range. Extensive tests are foreseen during the Computing and Detector Commissioning.

References

[1] ATLAS Collaboration, ATLAS High-Level Triggers, DAQ and DCS Technical Proposal, CERN/LHCC/2000-17, (2000)
[2] ATLAS Collaboration, ATLAS Computing Technical Design Report, CERN/LHCC/2005-022, (2005)
[3] A. Valassi, COOL web page, Available at http://lcgapp.cern.ch/project/CondDB/