The ATLAS High Level Trigger Configuration and Steering: Experience with the First 7 TeV Collision Data

Jörg Stelzer
on behalf of the ATLAS collaboration
Deutsches Elektronen-Synchrotron DESY, D-22603 Hamburg, Germany
E-mail: joerg.stelzer@cern.ch

Abstract. In March 2010 ATLAS saw the first proton-proton collisions at a center-of-mass energy of 7 TeV. Within the year, a collision rate of nearly 10 MHz was achieved. At ATLAS, events of potential physics interest are selected by a three-level trigger system, with a final recording rate of about 200 Hz. The first level (LVL1) is implemented in customized hardware, the two levels of the high level trigger (HLT) are software triggers. For the ATLAS physics program more than 500 trigger signatures are defined. The HLT Steering is responsible for testing each signature on each LVL1-accepted event, the test outcome is recorded with the event for later analysis. The steering code also ensures the independence of each signature test and an unbiased trigger decision. To minimize data readout and execution time, cached detector data and once-calculated trigger objects are reused to form the decision. In order to reduce the output rate and further limit the execution time, some signature tests are performed only on an already down scaled fraction of candidate events. For some of these signatures it is important for physics analysts to know the would-be decision on the event fraction that was not analysed due to the down-scaling. For this the HLT-Steering is equipped with a test-after-accept feature. The HLT-Steering receives the setup of the signatures from the trigger configuration system. This system dynamically provides the online setup for the LVL1 and HLT. It also archives the trigger configuration for analysis, which is crucial for understanding trigger efficiencies. We present the performance of the steering and configuration system during the first collisions and the expectations for the LHC phase 1.

1. Introduction

At the LHC design luminosity proton bunches will cross at a rate of 40 MHz. A highly selective and efficient trigger system identifies and accepts interesting physics events, while reducing the overall data rate to levels that fit the constraints of the data-flow and storage systems. The ATLAS trigger is a three-level system with a design final output rate of about 200 Hz [1].

The first level (LVL1) is implemented in custom-made electronics, reducing the event rate to less than 100 kHz. Within a latency of 2.5 μs, LVL1 takes a decision based on comparatively coarse objects which it receives from the calorimeter and the muon trigger electronics. This information is combined in the central trigger processor (CTP); the LVL1 accept decision is purely based on the multiplicities of these muon and calorimeter trigger objects. In addition the LVL1 can issue triggers on selected or random bunch-crossings. The High Level Trigger (HLT), which consists of the Level 2 (LVL2) and the Event Filter (EF) trigger, is implemented in software that is executed on a farm of about 2300 rack-mounted computing nodes. HLT algorithms have
access to data from all ATLAS sub-detectors. Executing speed-optimized algorithms the LVL2 analyses only data within $\eta \times \phi$ regions identified by LVL1 as being of interest. After a LVL2-accept signal the event building collects all detector and trigger data and passes it to the EF processes. At EF the same reconstruction algorithms as offline are executed, thereby using the full event data. Events accepted by the EF are written to mass storage. The LVL2 time budget is about 40 ms, the EF budget a few seconds.

During the data taking period of 2010 which lasted from the end of March through the middle of December, ATLAS recorded 45.0 pb$^{-1}$ of stable proton-proton collisions and 9.1 $\mu$b$^{-1}$ of stable heavy-ion collisions, with an average $pp$-data recording efficiency of 93.6%. The reliability and flexibility in operating the ATLAS HLT played an important role in achieving this impressive result. During the beginning of the year data was recorded based solely on the decision of the LVL1 trigger. The HLT started to actively reject events when the LHC had reached a luminosity of $2 \times 10^{30}$cm$^{-2}$s$^{-1}$, at the end of July 2010. From then to the end of $pp$ data taking in early November the LHC peak luminosity went up by a factor of 100, and the definition of the LVL1 and HLT lines and down-scaling factors, the prescales, had to undergo a rapid evolution. Furthermore, because of the drop in luminosity during the lifetime of an LHC fill, the trigger prescales have to be adjusted multiple times over the period of a run, in order to make use of the available data recording bandwidth.

Three key features of the ATLAS HLT are vital to achieve the robustness and flexibility desired for the trigger operation: the real-time performance monitoring and validation system, the flexible configuration system to adjust trigger prescales or even disable the trigger lines in case of problems, and the error catching and recovery system for failures in the HLT execution.

### 2. Requirements on the High Level Trigger

The HLT is designed in adherence to the following specifications:

- Ensure reliable and fast execution of trigger algorithms
- Ensure independent execution of all trigger lines
- Allow quick deployment of different trigger configurations to adjust to LHC beam conditions and ATLAS data taking needs

---

**Figure 1.** Design of the ATLAS Trigger.

**Figure 2.** Schematic layout of a trigger signature.
• Enable monitoring of trigger performance, trigger object properties, and execution timing
• Provide access to trigger decision, trigger objects, and trigger configuration for offline
   analysis

In the following more light is shed on the measures that were taken to fulfill the requirements
on robustness and timing. In the more detailed explanation of the trigger menu and the
HLT Steering framework that follows, the realization of trigger line independence, flexible
configuration, and trigger decision access will be made clear. A section on how this works in
practice comes thereafter. The last section will show the monitoring capabilities of the system.

2.1. Reliable and Fast Execution of Trigger Algorithms
The ATLAS HLT trigger community established a careful validation and deployment procedure
to guarantee a stable trigger execution. This resulted in robust trigger algorithms that are
protected against possibly missing or corrupted detector data, and perform well and within the
allowed time budget.

The algorithms are executed inside a framework, the HLT-Steering, that deals with
algorithmic errors and handles timeouts in case the processing time of the algorithm is
approaching a certain, configurable limit. In the rare case of a software crash or an uncaught
execution time-out, the data acquisition infrastructure takes care of restarting the HLT process.
In all cases the events are recorded in a special debug data stream for later investigation. This
guarantees a very robust HLT processing.

The HLT Steering implements the policy of early rejection during event processing. As soon as
it becomes apparent that a certain trigger line can not be satisfied, the remaining algorithms for
that trigger line are not executed. E.g. if for an electron trigger the energy of the electromagnetic
cluster is below the required threshold, the tracking algorithms for that electron candidate is
omitted. This approach minimizes the online event processing time, the data access rate and
the amount of retrieved data.

3. Trigger Menus and the Configuration and Steering Framework
The trigger menu is the incarnation of the ATLAS trigger program. In technical terms it is a
list of trigger signatures and corresponding prescaling values. A trigger signature is designed
to identify particle candidates, such as leptons and jets, or determine event properties, such
as total and missing transverse energy. A signature spans all three trigger levels, at each level
it consists of reconstruction and selection algorithms which are executed in order. Each level
has a corresponding prescale value, which can be used to control the accept rate of a trigger
signature or disable the trigger signature entirely. At the HLT the prescale decision is based on
the output of a pseudo random number generator. Each time a prescale of $p$ needs to be applied
to a signature, the prescale decision is positive if a randomly thrown number between 0 and 1 is
larger than $1/p$. Figure 2 shows an exemplary low threshold muon signature, detailing at Level
2 how each step consists of a list of reconstruction algorithms and a selection algorithm at the
end. If the selection is not passed then the processing of that signature stops. Another measure
to save CPU and data transfer resources is to apply the prescaling decision at the beginning
of the two levels of the HLT. In the multitude of trigger signatures, reconstruction algorithms
are frequently reused. By analysing the input of each algorithm the steering ensures that no
algorithm is executed twice on the same data. This is the third measure to minimize resource
usage.

The trigger configuration, which includes the trigger menu as well as algorithm properties
and further LVL1 hardware configuration parameters, is stored in the Trigger Configuration
Database (TriggerDB), a relational database schema running on Oracle servers. Once stored, a
trigger configuration is identifiable by three integer numbers, the trigger keys, which are used to
Figure 3. Example of LVL2 trigger prescale changes during a run. Here the L2_vtxbeamspot_activeTE_peb trigger is initially disabled, then enabled when LHC stable beams are declared. As the luminosity falls, the prescale factor is reduced, without restarting the run, in a few steps to keep the output rate between about 6 to 12 Hz.

retrieve 1) the configuration, 2) the LVL1 prescale values, and 3) the HLT prescale values from the configuration database. The TriggerDB provides a history of all trigger configurations ever used in ATLAS.

4. Trigger Operation
Preparing the HLT for data taking is a lengthy procedure that involves many tests of physics performance, output rates, and fault tolerance, drawn-out over many validation steps. Once a trigger menu is approved for data taking, the software release is installed on the online farm and the corresponding trigger configurations are stored in the TriggerDB. Typically one or two configurations exist per release. To operate the trigger at the different luminosities that can occur during a run, a larger number of prescale keys for LVL1 and HLT are provided with each configuration.

4.1. Updating the HLT Prescales During Data Taking
The ATLAS trigger system has been set up such that it is possible to change the prescale values during the run to any set of prescale values that exist in the Online TriggerDB. This is of importance not only for adjusting the prescales according to the LHC luminosity profile of a fill, but also for being able to react to unforeseen circumstances and turn off triggers if needed. The procedure of changing prescales adheres to the following requirements: no deadtime should be incurred, consistent prescales must be applied across the HLT farm, and everything must be reproducible offline.

The knowledge about the trigger prescales is vital for any data analysis, and it is necessary to have well-defined prescales for a given data period. ATLAS data is divided into runs and luminosity blocks (LB). During a LB, which is usually 2 minutes long, luminosity and conditions are approximately constant, the changing of prescales is permitted only between LB’s.

Prescales can be changed at LVL1 via the LVL1 Central Trigger Processor, or at the HLT, distributing the prescale set to all notes of the farm. Prescale update requests for the HLT are distributed with the data flow, events carry a flag that a prescale update is to be expected at a
certain LB. An HLT flag that is processing such a flagged event retrieves the new prescale set from the TriggerDB and keeps it in memory together with all previously used prescale sets. The access of the HLT notes to the TriggerDB happens through a hierarchical database proxy and caching system, reducing the impact on the database and the data flow to a negligible amount. Because the HLT node also stores a mapping between prescale sets and LB ranges the correct prescale set is always applied to any particular event. Any prescale change is also recorded in the ATLAS conditions database for use in physics analysis. Figure 3 shows an example of the use of prescale updates to keep the output rate constant during the run.

4.2. Updating the Online Beam Spot During Data Taking
A similar mechanism as the online prescales update is employed to update the beam-spot information of the HLT during the run. Knowledge of the beam-spot is important for the detection of displaced $B$-meson vertices. While the LHC beam-spot position is very stable, any unexpected change would have a negative impact on the $b$-trigger performance. Therefore the beam-spot position is constantly monitored. In case a large enough deviation from the currently used position is measured, the new position is, upon expert request, written to the conditions database. Via the event data flow the HLT nodes are notified of the update and forced to re-read the conditions database. The effect of the beam spot update can be seen in Fig. 4.

5. Monitoring the Trigger Performance and Resource Usage
Monitoring of the trigger rates and the resource usage of the trigger algorithms is naturally a task of the HLT Steering framework, since it controls the algorithm execution. In order to monitor the trigger rates the Steering simply collects the number of events passing each step of the trigger signatures. Together with the performance histograms of the trigger algorithms this information is collected through a hierarchical information gathering infrastructure and displayed in the trigger rates and performance monitoring applications.

Collecting information about the CPU usage and the rate and volume of data access of each algorithm is complicated by the fact that algorithm execution and data requests from the detector are cached. Therefore, on each event, all algorithm execution calls and data requests are monitored and, in case of a cached response, the time spent during the first and only execution is used for the accounting. Hence for each trigger signature the actual required time is measured. This method also provides all information needed to estimate trigger rates for new trigger menus and different luminosities. Figure 5 illustrates the capabilities of the ATLAS HLT resource usage monitoring.

**References**
[1] G. Aad et al. [ ATLAS Collaboration ], JINST 3 (2008) S08003.