The ATLAS Trigger Core Configuration and Execution System in Light of the ATLAS Upgrade for LHC Run 2

This content has been downloaded from IOPscience. Please scroll down to see the full text.
(http://iopscience.iop.org/1742-6596/664/8/082017)

View the table of contents for this issue, or go to the journal homepage for more

Download details:

IP Address: 137.138.124.206
This content was downloaded on 24/02/2016 at 12:28

Please note that terms and conditions apply.
The ATLAS Trigger Core Configuration and Execution System in Light of the ATLAS Upgrade for LHC Run 2

Lukas Heinrich, on behalf of the ATLAS TDAQ collaboration [1]
New York University, New York, USA
E-mail: lukas.heinrich@cern.ch

Abstract. During the 2013/14 shutdown of the Large Hadron Collider (LHC) the ATLAS first level trigger (L1) and the data acquisition system (DAQ) were substantially upgraded to cope with the increase in luminosity and collision multiplicity, expected to be delivered by the LHC in 2015. Upgrades were performed at both the L1 stage and the single combined subsequent high level trigger (HLT) stage that has been introduced to replace the two-tiered HLT stage used from 2009 to 2012 (Run 1). Because of these changes, the HLT execution framework and the trigger configuration system had to be upgraded. Also, tools and data content were adapted to the new ATLAS analysis model.

1. Introduction
The ATLAS experiment is one of four major experiments at the Large Hadron Collider (LHC) [2] at CERN. The ATLAS detector [3] relies on high-precision tracking and calorimetry sub-detectors with many million read-out channels to precisely capture collision events. Due to constraints both in the bandwidth of the read-out system as well as the permanent (offline) storage capacity, it is not feasible to record all collisions, most of which are from well-understood physics processes. Hence, ATLAS employs a multi-staged trigger system to select events of particular interest.

During the first run of the LHC from 2009–2012 (Run 1), the LHC provided proton-proton collisions at a rate of 20 MHz and the ATLAS trigger was configured as a three-stage system. The first stage, the Level-1 trigger (L1), was implemented using custom electronics and read out low-granularity calorimeter and muon spectrometer data. It reached a decision within 2.5 µs to reduce the rate to 75 kHz. In addition, L1 also identified the location in the detector (Regions-of-Interest, RoIs), where the interesting activity occurred that led to accepting the event. The RoIs guided the second stage, the Level-2 trigger (L2), which read out full-granularity data only from the RoIs to perform a partial reconstruction and further reduce the event rate to about 4 kHz within 40 ms. In case of acceptance by L2, the entire event data was read out by the Event Filter (EF), and a final decision on the basis of a fully assembled event was made within 4 s. The L2 and EF stages, which ran on a commodity PC farm, were also collectively referred to as the High Level Trigger (HLT) and used similar software as the offline reconstruction.

After a two-year-long shutdown (LS1) in 2013–2014, the LHC will resume operation in summer 2015 at an increased center-of-mass energy of 13 TeV and increased instantaneous
luminosity of up to $1.6 \times 10^{34}$ cm$^{-2}$ s$^{-1}$. The resulting higher trigger rates and event data sizes, the latter stemming from a larger number of simultaneous proton-proton collisions referred to as pileup, pose a challenge to the trigger and data acquisition system (TDAQ) of the experiment, which operated in some areas well beyond its design values already in Run 1. The design of the trigger for Run 2 was simplified to a two-stage system by merging L2 and EF into a unified HLT stage. Basic figures of merit for this system are a L1 output rate of 100 kHz and a HLT event writing rate of 1 kHz. During the shutdown a number of upgrades to the TDAQ system have been undertaken to meet these new requirements, a selection of which is presented in this document. A schematic view of the Run 2 trigger system is shown in Figure 1.

2. Trigger Hardware Upgrades

2.1. Level 1 Trigger

For Run 2 the Level 1 trigger design has been extended by a topological trigger module (L1Topo) [4]. While the L1 strategy in Run 1 could only use the multiplicity of candidate trigger objects identified by the Level 1 calorimeter and muon trigger hardware, this module is capable of selecting events based on topological relationships between these candidate objects. Interesting selection criteria include the invariant mass of multiple trigger objects, the scalar transverse momentum sum and angles between L1 trigger objects. The module receives RoI data at a rate of 1 Tb/s which is then processed in less than 100 ns using algorithmic firmware, which is loaded onto on-board FPGAs. Data indicating which algorithms have passed are then passed on to the Central Trigger Processor (CTP).

The increase in pileup in Run 2 will result in higher occupancies in the calorimeter system. Consequently, the L1 calorimeter trigger (L1Calo) [5] has been upgraded with a new pre-processing module. The new module will be capable of correcting for pileup using a dynamic pedestal correction. Additionally, the output data format has been extended from simple hit
counts to more descriptive Trigger Objects (TOBs) which provide candidate $\eta$, $\phi$, and $E_T$ information to the topological trigger. Similarly the Level 1 muon trigger (L1Muon) received a firmware upgrade to send coarse $\eta$, $\phi$ and $p_T$ information to the L1Topo modules.\footnote{In the ATLAS coordinate system $\eta$ and $\phi$ are the pseudo-rapidity and polar angle respectively while, in the limit of massless particles, the ‘transverse energy’ $E_T$ corresponds to the momentum component transverse to the beam axis, which is denoted $p_T$.}

The central trigger processor (CTP) \cite{6} receives inputs from all other L1 components and ultimately decides whether an event passes the first trigger stage. The internal bus has been overclocked to allow twice as many thresholds from the L1Calo system to be included in the selection. The L1 decision is made using a number of predefined trigger items, logical combinations of the input signals that describe criteria for accepting events such as thresholds, multiplicities, and flags set by the topological trigger. For Run 2 the number of trigger items has been increased from 256 to 512 to allow a more refined selection. The CTP has also been upgraded to receive inputs from the new L1Topo module. Also, it can now be run with up to three partitions, concurrently running instances of the CTP software. Only one of the partitions is interfaced to the HLT and DAQ systems, while the other two are intended for commissioning and calibration. To support this partitioning, a new control software architecture was developed.

2.2. High Level Trigger

The high level trigger and data acquisition system have been thoroughly upgraded for Run 2. Most notably, the L2 and EF stages were merged into a single HLT stage resulting in greater flexibility on how the event read-out and triggering is organized. During Run 1, resources in the HLT computer farm have been dedicated specifically to either the L2 or the EF stage. In Run 2, all computing resources will act as unified HLT nodes that will execute both the RoI-based limited read-out decision making as well as the full event assembly and decision. In such a scheme the event-building stage can be scheduled dynamically at any point in the data processing as shown in Figure 3(a). During Run 1 a stringent limiting factor was the rate of data-requests from processing nodes to the read-out system PCs (ROS). First the Level 2 process would request full-granularity data from the RoIs and after L2 acceptance the event-building process would request the entire event data (including the RoI data already requested by L2). With merged HLT processing nodes, data needs to be requested only once from the ROS hence saving network bandwidth and decreasing the ROS data request rate. Finally, by having only a single kind of node, the load-balancing capabilities of the computer farm are significantly improved.

The network providing the nodes with the read-out data has been restructured to reflect the merged HLT design. In Run 1 each ROS had links into two separate networks. The data collection network provided the L2 processing units with the read-out data. Upon accept the entire event was read out by dedicated event building nodes and the fully built events were distributed via a second back-end network to the EF nodes. For Run 2, the design was simplified into a single network with a 6 Tb/s bandwidth. A new generation of ROS PC was equipped with two 10 Gb/s links into the data collection network via which the HLT computing nodes can request event data. Also, the new ROS machines now hold new read-out buffer input cards (ROBIN) which will be able to sustain higher access rates.

3. Configuration and Execution Model

3.1. Trigger Configuration

The trigger configuration is a system to describe both the hardware and software trigger components. The trigger menu is a high level description of the physics signatures that are to be recorded and is compiled in close collaboration with the physics working groups. The menu relates physics objects and multiplicities to specific algorithms in the trigger software. For

\textbf{21st International Conference on Computing in High Energy and Nuclear Physics (CHEP2015) IOP Publishing}
the L1 stage, the menu lists the 512 trigger items that have been built from the L1 trigger objects as well as the L1Topo algorithms. For the HLT the menu is organized into approximately 2000 trigger chains (twice the Run 1 value) that each describe a sequence of algorithms that need to be executed in order to test for a certain physics signature. HLT algorithms are classified as being either of feature extracting (FEX) or hypothesis testing (HYPO) type. While the former attempt to reconstruct physical objects such as tracks or calorimeter energy clusters, the latter evaluate the quality of the reconstructed object to mark it as satisfying the chain’s selection criterion. In an electron chain, for example, FEX algorithms might reconstruct track and calorimeter clusters, while a subsequent HYPO algorithms might check if the cluster and track are consistent with the electron hypothesis based on e.g. the cluster shape or transition radiation detected in the tracker. For certain chains, the trigger rate can be too high to write every passing event to disk, therefore a prescale factor $p$ may be applied so that a passing event is recorded only with a probability of $1/p$ reducing the output rate by a factor of $p$. As prescaling is applied before the algorithms are executed, the execution time is also lowered.

Figure 2: Excerpt of the trigger database schema.

The trigger configurations for L1 and HLT are stored in a relational Oracle database (TrigDb) which can be viewed and modified via the TriggerTool, a graphical user interface. For Run 2 the database schema, shown in Figure 2, has been amongst other things upgraded to incorporate the L1Topo configuration. Also, the architecture and interface to the TriggerTool have been significantly improved. Based on the database, any trigger configuration can be uniquely identified by three keys. The supermaster key (SMK) uniquely identifies the menu, while the L1 and HLT prescale keys (L1PSK/HLTPSK) identify the prescale sets for their respective trigger stage. While each ATLAS run (typically corresponding to one LHC proton beam fill) uses a single SMK, multiple prescale keys are applied to optimize bandwidth usage as the beam intensity drops over the course of the run. In Run 2 the prescale application process will be simplified and automated in order to maximize the data taking efficiency.

3.2. HLT Steering

When processed by the HLT, the execution of the algorithms in the correct order is driven by the HLT Steering software framework. After applying the prescales for the event at hand, the remaining chains are evaluated in a data-driven manner. Starting from the L1 items as seeds, for each chain the next available step is executed. A step consists of a list of FEX algorithms and a final HYPO algorithm and is marked as passing based on the latter’s response. As soon as a step of a chain fails, that chain is marked as not passing and is not further considered. Chains for which all steps have succeeded are marked as passed and the event is recorded in the case of one or more passing chains. Several chains may share individual steps and the steering system caches results in order to avoid multiple passes over the same data. The first chain passing also triggers the retrieval of all remaining data from the ROSes. Upon acceptance of the event by the HLT, the trigger information including the decision as well as all objects reconstructed by
the trigger algorithms are serialized into binary form and included in the event data which is written to offline storage. For Run 2 the HLT algorithms have been rewritten to make use of the merged HLT design. Generally, L2-type algorithms requesting only partial full-granularity along with more comprehensive EF-type algorithms still exist. The corresponding steps, however, are now considered part of the same chain.

3.3. Multiprocessing at the HLT

With clock speeds of processors stagnating in recent years, computational power has been gained mostly from increasing the number of cores on a single die. Many-core processors such as new generations of Intel’s Xeon family of processors, which the HLT uses, already have up to 24 real and 48 virtual cores. Volatile memory capacities, however, have not been increasing at a similar rate. This poses a challenge to the execution model in the HLT which has relied on running multiple independent processes each with their own memory address space to exploit the fact that every event can be processed independently — largely, the HLT is an embarrassingly parallel problem. In this scheme the total amount of memory is the limiting factor in determining the number of concurrent processes that can be run on a machine each of which would require around 2 GB.

Fortunately much of the memory needed by the HLT processes is not unique to the event. Examples of such common data are the magnetic field maps, the detector geometry, and run conditions data. The amount of data unique to the event is a moderate 300 MB. To make optimal use of this, ATLAS has reworked its multiprocessing setup for Run 2 to make use of the copy-on-write feature of the Linux kernel which allows forked processes to use common memory as long as they do not modify it (when they do, the modified memory pages are copied and only then use additional memory). In the implementation at the HLT, shown in Figure 3(b), a single process, HLT\textsubscript{0}, will be started and undergo various initialization procedures to set up the shared data. After initialization, the process will fork into many copies of itself (HLT\textsubscript{1} \ldots HLT\textsubscript{N}). All these processes may now share the common memory and will then independently process events. Using this feature, the memory available in the node is utilized in a much more efficient manner allowing an increased number of HLT processes to run on a single node.

3.4. Data Scouting

The reconstruction algorithms running in the HLT are very similar to the offline reconstruction algorithms used to process the events once they are written to disk. In fact, in many cases a large amount of code is shared between online and offline algorithms. Consequently, the efficiency and resolution with which physics objects such as jets, electrons and muons are reconstructed online is almost as good as when done offline. This opens up an intriguing opportunity for triggers
with inhibitive large rates and hence large prescale factors.

 Normally, an HLT-accepted event is classified into one or more data streams and the entire event data is written to disk in its raw form awaiting offline reconstruction. For Run 2, a new data stream type, the data scouting stream, has been added. Instead of recording the entire event data, in this stream only collections of the physics objects reconstructed by the trigger are written to disk. Writing only a fraction of the event data enables the experiment to run high-rate triggers in an unprescaled configuration. The data-scouting streams will provide high-statistics samples that can be used both for calibration purposes as well as certain searches for new physics.

4. Trigger Analysis and Monitoring

4.1. Offline Analysis Tools

An essential component of the core trigger software is to provide tools to access the trigger information associated to reconstructed offline data. With the increased instantaneous luminosity in Run 2 it is expected that ATLAS will record data of the order of 100 fb\(^{-1}\) per year. To prepare for these unprecedented data sizes, the event data model (EDM) of the offline software has been adapted. The main objective of the new EDM was to make the output data format of the reconstruction, the \(z\text{AOD}\), natively readable in the data analysis framework ROOT [7]. As this was not possible during Run 1, ROOT-readable derivative datasets were produced using significant amounts of computing and storage resources which would not be feasible in Run 2. The trigger is the only online software component that has been affected by the EDM change, since the HLT reconstruction algorithms also use the offline EDM. Therefore, the online software was adapted in order to enable serialization of \(z\text{AOD}\) data.

For offline analysis, the main access to trigger information is provided by the TrigDecisionTool. This tool has been adapted to enable use in both the reconstruction software framework (Athena) as well as a ROOT-only analysis environment.

4.2. Trigger Cost Monitoring

To ensure an efficient operation of the trigger, the computational cost of the trigger system must be well understood. Characteristics such as execution time or the number of data requests are crucial pieces of information when preparing new trigger configurations, during data-taking as well as for later analysis. The Trigger Cost Monitoring framework was developed to record this information for the entire trigger system and calculate the cost of each trigger alone and in combination with others, taking into account data access and algorithm execution caching. A web-based interface, shown in Figure 4, is available to quickly access the results of the cost calculation. An important use case for the cost monitoring is the prediction for the rates of individual trigger chains which in turn can inform the composition of the trigger menu. For Run 2 the cost monitoring has also been updated to reflect the merged HLT design.

5. Conclusion

The trigger of the ATLAS experiment is a crucial component to select the most interesting collision events provided by the LHC in light of limited bandwidth and storage capacities. It has performed with remarkable success during the first round of data-taking from 2009 to 2012. To ensure a similar performance under the even more challenging conditions of Run 2, the system has been thoroughly upgraded. The L1 trigger now is able to select a larger variety of physics signatures on the basis of a wider set of characteristics, notably topological information. The HLT architecture has been simplified to a single streamlined trigger stage. This change has also been reflected in a new and improved data acquisition system ready to sustain higher access rates and larger bandwidth demands. The online trigger configuration and execution software was rewritten to adapt to and make use of the merged design. Finally, the offline software
Due to bandwidth and storage limitations not all proton collisions can be stored permanently for analysis. After the first Long Shutdown (LS1), LHC Run 2 and trigger analysis tools have been updated. The trigger event data model (EDM) now supports the new ATLAS event data model and data analysis outside of the reconstruction software framework.

References
[2] Evans L and Bryant P 2008 JINST 3 S08001