Operational experience with the new ATLAS HLT framework for LHC Run 3

Aleksandra Poreba[1,2,*] *on behalf of the ATLAS Collaboration*

[1]CERN
[2]Ruprecht-Karls-Universität Heidelberg

Abstract. Athena is the software framework used in the ATLAS experiment throughout the data processing path, from the software trigger system through offline event reconstruction to physics analysis. For Run 3 data taking (which started in 2022) the framework has been reimplemented into a multi-threaded framework. The ATLAS High Level Trigger (HLT) system has also been updated to rely to a greater extent than in Run 2 (data taking between 2015-2018) on common solutions between online and offline software. We present the now operational new HLT system, report on how the system was tested, commissioned and optimised. In addition, we show developments that have been made in tools that are used to monitor and configure the HLT, some of which are designed from scratch for Run 3.

1 Introduction

The ATLAS detector [1] is one of four particle physics detectors at the Large Hadron Collider (LHC), designed for Standard Model measurements and searches for new particles. The experiment consists of multiple layers of dedicated subdetectors, including tracking detectors and calorimeters. The collisions are delivered to the ATLAS interaction point with a rate up to 40 MHz for proton-proton data taking.

Not all of the collisions recorded by the ATLAS detector are saved to the output data *streams*, as the amount of the data is too large to be saved and stored using today's available technologies, moreover, not all of them are relevant for physics analysis. The ATLAS Trigger system aims to filter the detector output in real time based on predefined criteria. It consists of two levels: the Level 1 (L1), a hardware trigger applying a coarse selection, and the High Level Trigger (HLT), a software trigger tightening the selection based on the L1 decision and the detector readout.

The HLT event selection is achieved by a set of algorithms performing event reconstruction and selection based on the reconstructed features. The algorithms are organized in sequences, focusing on different event features. The sequences build the selection *chains*, which are organized in the selection *menu*. Additionally, each chain has an associated *prescale* factor, reducing the number of events in which it will be executed, and limiting the HLT online output rate for nonessential selections.

*e-mail: aleksandra.poreba@cern.ch

The order of steps within a chain is important - algorithms running early within a chain should reject as many events as possible, so the more CPU-intensive algorithms are executed only on the smallest subset of events possible. This design is called the *early rejection mechanism* and it was applied in the HLT event selection in order to save CPU resources.

The online event selection code is part of the Athena framework [2], shared with offline reconstruction, simulation and physics analysis. The events are processed on a computing farm with approximately 60,000 real CPU cores (2023).

2 Run 3 commissioning

2.1 Multi-Threaded Framework

The first attempts to parallelize the ATLAS event selection have already been used in Run 2, by implementing a Multi-Processing approach. Online, the Athena processes were managed by the HLT Processing Unit (HLTMPPU), where event selection child processes (worker processes) were forked from a mother process. The mother process itself was not participating in the event selection processing [3], but was used to monitor the children and served as a template for new forks when needed. For the offline reconstruction and simulation, AthenaMP [4] was prepared. In both cases, memory usage is reduced by relying on read-only memory sharing (e.g. magnetic field map, detector geometry) via the *copy-on-write* mechanism, while threads share both read and write memory.

Even though AthenaMP reduced the memory consumption for Run 2 offline reconstruction, further gains could not be achieved with the Multi-Processing approach. The plan for Run 3 included the implementation of a Multi-Threaded framework as well as sharing the same code between online and offline to simplify the maintenance of the particle reconstruction code [5]. Therefore, the online event selection code had to support the Multi-Threaded mode in order to reduce the memory footprint, even though it was not an issue for online operation. Apart from general improvements related to the redesign of the code, the upgrade simplified the possible integration of computing accelerators for future running periods.

With the new Multi-Threaded framework, the configuration of the HLT Processing Unit and its CPU resource utilization is defined by three parameters:

- Number of forked worker processes
- Number of threads within the worker processes
- Number of event slots, defining how many events can be executed in parallel per worker process

To maximize the physics output, the best resource utilization configuration was chosen supported by many studies. The results in Figure 1 were collected by measuring four different ways of parallelisms: pure Multi-Threaded, pure Multi-Processing, and two hybrid configurations. They were performed in a local environment using a machine identical to those used in the ATLAS HLT computing farm during data-taking.

The best performance in 2022 was achieved with a pure Multi-Processing configuration. During the standalone studies, the results are presented in Figure 1, as well as during the data taking, it showed the highest event throughput. The memory and CPU usage is worse in comparison to the pure Multi-Threaded or hybrid modes but these limitations do not affect the online operation.

A pure Multi-Threaded configuration with lower event throughput is still used for Monte Carlo simulation production, where memory savings are necessary. Hybrid configurations were also considered for online event selection, giving similar gains in memory usage without a throughput penalty. In 2023, the hybrid configuration with 100% overallocation (the

number of events processed in parallel is 100% higher than the number of CPU cores) was used to lower memory usage.

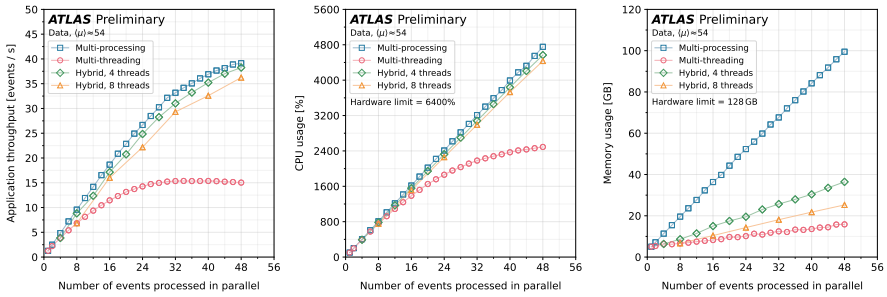

Figure 1. (Left) Application throughput in events / s, (center) CPU usage (CPU time divided by wall time) in percent and (right) memory usage in GB as a function of the number of events processed in parallel for the ATLAS Athena application executing trigger selection algorithms. The measurements were performed with a data sample containing a mix of events representative of the real ATLAS HLT input data and trigger selection configuration identical to one used during data-taking. [6]

2.2 HLT Configuration

Apart from the software improvements, the way of storing and accessing the HLT configuration was redesigned as well. For Run 3, it is saved in JSON format, compared to the XML format which was used for Run 2. The JSON configuration files can be stored as Binary Large Objects (BLOBs) in a relational database. The HLT configuration information can be provided transparently to the HLT applications in different ways: from a database (used during operation), from JSON files, from a configuration description in Python (used for standalone studies) or from 'in-file meta-data' (mostly used for offline reconstruction). The unification of the configuration format between different workflows simplifies the user's experience and the developer maintenance.

For the data taking, many sets of configuration files with different prescale values are prepared, in order to adapt to the current luminosity level throughout a run. As the instantaneous luminosity decreases during a run, more resources (CPU, bandwidth) are available, therefore, the prescale factors can be adjusted to maximize the physics output. The prescale values are based on preliminary performance studies of the selection's cost. Some of the selection chains are enabled only at the end of the run when the resource usage is low enough. The configuration changes are visible in the recorded rates of output streams: an example of the recorded output rate of HLT streams is presented in Figure 2.

The HLT configuration stored in the database is accessible via TriggerToolWeb, a modern web-based application, which was developed for Run 3 to replace a legacy Java application [8] . It is widely used during online operation to display, modify, and compare the menu configuration sets with different chains and their prescales.

2.3 Hardware upgrades

Not only the Trigger software was improved in 2022, but also the HLT farm was upgraded with new dual processor servers with AMD EPYC 7302 CPUs (sixteen real cores with two

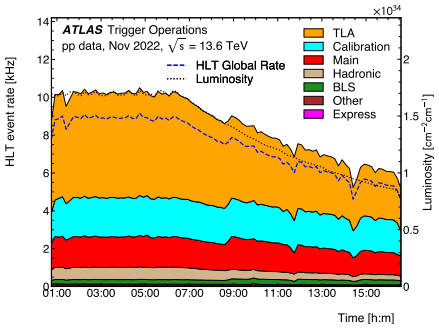

Figure 2. The rate output to the HLT streams in a 2022 proton-proton run. The total HLT event rate is lower than the sum of the stream rates, because the same events may be written to multiple streams. Periodic increases in the rate and bandwidth of support triggers are caused by prescale changes towards the end of the run as the luminosity and corresponding overall resource usage decline. [7]

hyper-threads per core). The total farm performance improved from 1.2×10^6 HS06[1] at the end of Run 2 to 1.7×10^6 HS06 in 2022 with 60% of the racks being replaced and 2.0×10^6 HS06 in 2023. The upgrade was done gradually, therefore different CPU rack configurations had to be used for the old and new types of machines.

Apart from the HLT farm CPUs, the Read-Out machines were planned to be updated but unfortunately, due to availability issues, the upgrade was delayed to the 2022/2023 winter technical stop. To mitigate the occurring issues with the performance of the readout machines, a *prefetching mechanism* was implemented for the new framework. At the beginning of a selection step, a special 'InputMaker algorithm' requests all necessary detector data in a RoI[2] to perform reconstruction and to reduce the frequency of data requests.

3 Performance monitoring tools

3.1 Online Cost Monitoring

The Cost Monitoring [10] summarizes the resource usage of HLT algorithms, HLT selection steps and chains and also the resource impact of HLT data requests. It consists of information in the form of ROOT [11] histograms and CSV tables, including the mean event processing time, the algorithm execution time per event, and the readout retrieval time. The Cost Monitoring data are collected in parallel to physics data taking and, after post-processing, the results are automatically published on a dedicated website. The Cost Monitoring was updated to support the Multi-Threaded framework for Run 3 [12].

Studies based on the Cost Monitoring data include analysis of the HLT processing time of one event as a function of pile-up[3]. An example of the results is illustrated in Figure 3. Based on the distribution, future resource needs can be estimated. The mean HLT Processing time decreases with decreasing average pile-up due to a reduction in event complexity. The changes in the HLT configuration are visible in the distribution, including the enabling of additional triggers when instantaneous luminosity falls approximately below 1.5×10^{34} cm^{-2} s^{-1} during a run.

[1] HS06 - HEP-SPEC06 benchmark [9], a measure of the CPU performance.

[2] RoI - *Region of Interest* is an area in the detector where candidates for particles were identified by L1 Trigger and are passed to the HLT for further analysis during the online data taking.

[3] Pile-up - number of particle interactions per bunch crossing

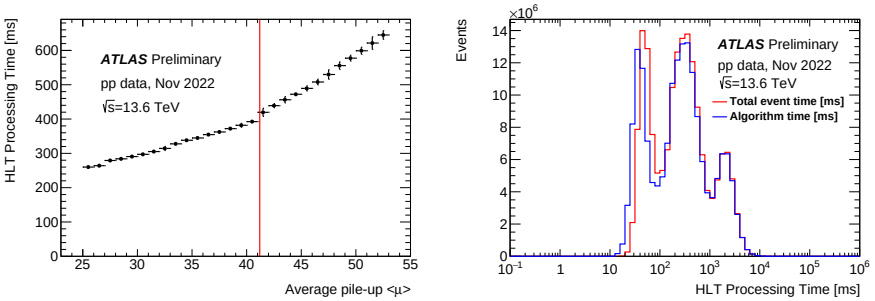

Figure 3. (Left) Mean HLT Event Processing time as a function of the average pile-up in a 2022 proton-proton run. The vertical line marks the enabling of additional trigger selections. Error bars denote the Gaussian width of the underlying per event measurements. (Right) An example of the HLT Processing time distribution per event in a 2022 proton-proton run for an instantaneous luminosity of 1.8×10^{34} cm^{-2} s^{-1}. The total event time includes algorithm execution time and time spent on framework operations (including algorithms scheduling and data traffic). [7]

The analysis of the online performance of algorithms helps to identify and improve the sub-optimal areas. The Figure 3 (right) illustrates the distribution of execution time of algorithms as well as total event time, including algorithm execution time and time spent on framework operations (i.e. including algorithm scheduling and data traffic). The latter takes a visible fraction of short events and is unnoticeable in events with execution times longer than 100 ms. In Figure 3 (right), three peaks can be identified, representing fast (approximately 30 ms), medium (approximately 300 ms), and slow (approximately 2s) events. The last type of the events is the rarest due to the early rejection mechanism.

3.2 Trigger Rate Presenter

The Cost Monitoring data is usually available only 48 hours after the end of the run due to the need for additional processing. During the data taking, immediate monitoring is necessary in order to observe any performance variations, occurring e.g. due to detector malfunctioning or change of beam conditions.

The Trigger Rate Presenter (TRP) is publishing the current trigger rates along with diagnostic information including the memory usage or performance of readout systems. The tool was already available in previous data-taking periods as a standalone GUI application, however, it was updated for Run 3 to use a web-based display and compatible publishing.

The results from TRP are displayed on web-based Grafana dashboards, allowing the user to adjust the monitoring time frame and the form of values plots or tables. The data is archived with the P-BEAST [13] data storage service and accessible for later analysis. For Run 3, the web display includes the rates of individual HLT and L1 triggers.

4 Conclusions

The ATLAS HLT was vastly improved in preparation for Run 3, including a redesign of the HLT framework to support the Multi-Threaded mode and to share the vast amount of reconstruction modules with offline particle reconstruction. Apart from the software improvements, the hardware was upgraded as well, including the HLT farm upgrade increasing the performance to 2.0M HS06 (start of 2023). However, the performance was limited by the

read out system bottlenecks, that couldn't be upgraded in time due to availability issues. The bottlenecks were mitigated by data prefetching techniques and were resolved by the upgrade of the read out hardware in 2023.

To assess the HLT online performance, many tools were prepared to provide an overview of HLT performance on different levels, including physics signature selection, HLT algorithms, and global CPU resource needs. Based on the monitoring outcome, the HLT, (i.e. algorithms, and configuration), was optimized to maximize the physics output.

Acknowledgement

This work has been sponsored by the Wolfgang Gentner Programme of the German Federal Ministry of Education and Research (grant no. 13E18CHA)

References

[1] The ATLAS Collaboration et al, JINST **3**, S08003 (2008)

[2] ATLAS Collaboration Athena (21.0.127) Zenodo (2021)

[3] Rafal Bielski and on behalf of the ATLAS Collaboration, J. Phys.: Conf. Ser. **1525**, 012031 (2020)

[4] Paolo Calafiura et al, J. Phys.: Conf. Ser. **664** 072050 (2015)

[5] Charles Leggett et al, J. Phys.: Conf. Ser. **898**, 042009 (2017)

[6] Trigger Core Software Public Results
https://twiki.cern.ch/twiki/bin/view/AtlasPublic/TriggerCoreSWPublicResults

[7] Trigger Operation Public Results
https://twiki.cern.ch/twiki/bin/view/AtlasPublic/TriggerOperationPublicResults

[8] Carlos Chavez et al, J. Phys.: Conf. Ser. **664** 082030 (2015)

[9] Michele Michelotto et al, J. Phys.: Conf. Ser. **219** 052009 (2010)

[10] Tim Martin and on behalf of the ATLAS Collaboration, J. Phys.: Conf. Ser., **898**, 032007 (2017)

[11] Rene Brun and Fons Rademakers, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997)

[12] Aleksandra Poreba and on behalf of the ATLAS Collaboration, COMPUTING AND INFORMATICS, **40**, 4 (2021)

[13] M. -E. Vasile, G. Avolio and I. Soloviev, IEEE Transactions on Nuclear Science, **70**, 6 (2023)