Summer Student Programme 2023

Utilisation of GPUs for the ATLAS trigger software and implementation of machine-learning algorithms for muon reconstruction in the ATLAS High-Level Trigger

Alina Mariana Soflau

Supervisors: Francesco Giuli, John Kenneth Anders

Abstract. The large increase in luminosity planned for the High Luminosity LHC gives rise to many challenges for the trigger and data acquisition system. An instantaneous luminosity up to $7 \times 10^{34} cm^{-2} s^{-1}$ is expected, corresponding to an average number of inelastic pp collisions per bunch crossing of around 200. For such data taking conditions, due to the unprecedented number of particle hits in the tracker system, ATLAS is putting effort to include multithread computation devices in the trigger architecture. Graphics Processing Units are one of the technologies under investigation. Moreover, track reconstruction algorithms must have stable performance with respect to pileup, in order to ensure the computing requirements of the experiment. In this note a study of the performance of a fast track reconstruction algorithm with the new ATLAS Inner Traker geometry is presented.

1 A new Inner Tracker Detector

ATLAS has an ambitious Phase-II detector upgrade programme in preparation for High Luminosity LHC (HL-LHC) [1] in view of the expected rise in instantaneous luminosity and consequently, in the number of simultaneous pp interactions. A central part of this programme is the replacement of the current Inner Detector (ID) with a new Inner Tracker (ITk) [2]. The ITk was designed to provide a traking performance up to an average number of inelastic pp collisions per bunch crossing (*pile-up*) of around $\langle\mu\rangle = 200$, in a pseudorapidity range of $|\eta| < 4$. The Inner Detector for comparison covers a range of $|\eta| < 2.5$ and operates approximately at $\langle\mu\rangle = 34$ (average value of pile-up for Run 2 data taking).
A schematic view of one quadrant of the ITk is shown in Figure 1. The detector has 13 m^2 of pixel detectors with 5 billion readout channels and 160 m^2 of strip detectors with 50 million read-out channels. It consists of a strip subsystem and a pixel subsystem.

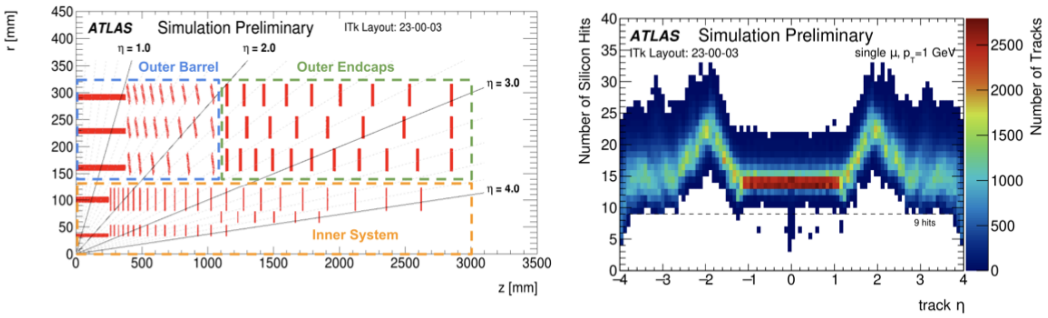

Figure 1: A schematic layout of one quadrant of the ITk pixel system is shown in the left panel. On the right panel, are displayed the simulated hits in silicon pixel layers of the ITk.

Figure 2 shows an exponential increase of the reconstruction time as a function of pile-up. This poses a significant challenge for the track reconstruction and its associated computing requirements due to the unprecedented number of particle hits in the tracker system. Dedicated algorithms have been developed to speed up the track reconstruction and further optimise the default algorithms of the ATLAS detector.

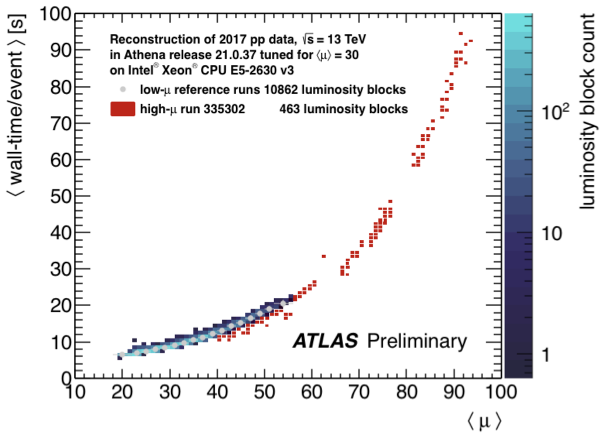

Figure 2: The reconstruction time is increasing exponentially as a function of pile-up. Significant improvements will be required to increase the speed of the algorithms that perform tracking as well as those that process information from the other detector systems [3].

2 Fast Track Finder

Track reconstruction of particles is a complex process, due to both the density of each event in the detector as well the helical trajectory the particles take due to the solenoidal magnetic field. Tracks are described by five parameters with respect to the beamspot position: d_0 (the transverse impact parameter), z_0 (the longitudinal impact parameter), ϕ (the azimuthal angle of the track momentum), ϑ (the polar angle of the track momentum), and q/p (the ratio of the track's charge and momentum).
Fast Track Finder (FTF) [4] is an algorithm developed for Run 2 in order to create medium-quality track candidates as quickly as possible, early in the High Level Trigger (HLT) system. Its design philosophy is to prioritise efficiency over purity. In this section a general overview of how Fast Track Finder algorithm operates is presented.

- The input of HLT is in the form of Regions of Interest (RoIs). They are geometrical regions of the detector that have been flagged by the Level-1 as those containing the interesting physics candidates. The algorithm initiates by searching for sets of three closely space points within these RoIs (*track seeds*).

- Once triplets are identified, Kalman filter algorithm is employed to extend these seeds and build more complex track candidates, by extrapolating the trajectories of the particles through the detector layers.

- An ambiguity scoring mechanism is employed in order to remove duplicate tracks, ensuring that each particle is represented by a single reconstructed track. Beneficial or penalty track scores are assigned, based on various track characteristics. Each hit associated with the track contributes to a higher score, favoring fully reconstructed tracks over small track segments. Additionally, measurements from different sub-detectors are weighted differently, giving preference to precision measurements, such as pixel clusters, while downgrading measurements from less precise detector components.

- An additional filtering is applied to higher quality tracks identified through the previous steps. They undergo further refinement using a fast Kalman filter track fitter.

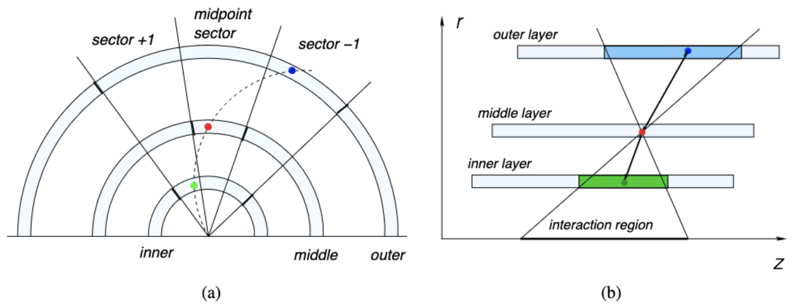

Figure 3: Schematic illustration of track seed formation. The search is performed within specific bins of r (radius) and approximately 50 different sectors in ϕ. Formation of a track seed begins from a central point (the midpoint of a triplet) and selects outer and inner points at larger and smaller radii, respectively, relative to the central point.

The tracks identified after this first fast tracking stage are used to seed the precision tracking stage (PT) to reduce CPU usage. While the fast tracking is based on trigger-specific pattern recognition algorithms, the precision tracking is based on offline tracking algorithms.

3 FTF with ITk geometry: results and discussion

For this study, FTF algorithm was implemented to work with an ITk dataset for the Phase-II Upgrade. More precisely, FTF was run as an offline algorithm: this allows for FTF to run as FullScan (uses the entire detector region), but does not allow use of RoIs created by the trigger.
The primary goal is the evaluation the algorithm's CPU efficiency in reconstructing muon tracks, while varying different parameters within the FTF, at $\langle\mu\rangle = 200$. This analysis is based on simulated pp collisions events with $\sqrt{s} = 14$ TeV. Three distinct datasets are separately analysed: simulated single muon events with a p_{T} of 50 and 100 GeV, as well as single lepton events from $t\bar{t}$ pairs productions. The particles considered must satisfy $|d_0^{truth}| < 2$ mm and $|z_0^{truth}| < 200$ mm.

3.1 Single Muon Sample with 50 GeV

Events containing single muons with constant transverse momentum of 50 GeV have been considered. All tracks are required to be within the fiducial region defined by $p_{\mathrm{T}}^{track} > 0.5$ GeV and $|\eta| < 4$. The required number of hits per track is *at least* 7 for the entire range of $|\eta| < 4$.

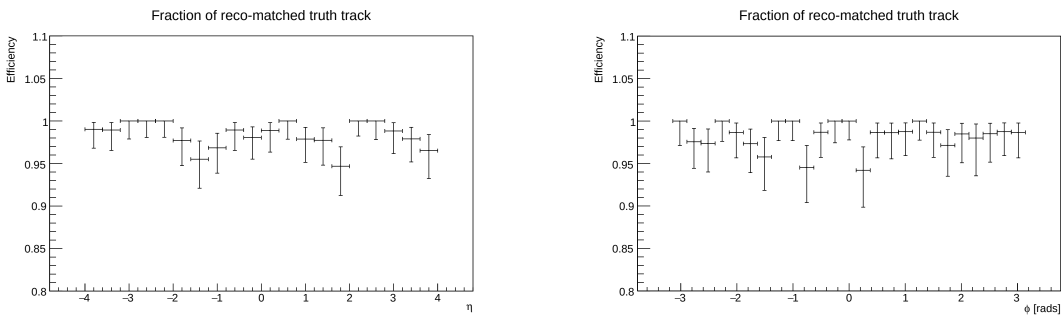

Figure 4: Track reconstruction efficiency versus η (left) and versus ϕ (right) are shown. The uniform behaviour as function of the azimuthal angle reflects the symmetrical design of detector.

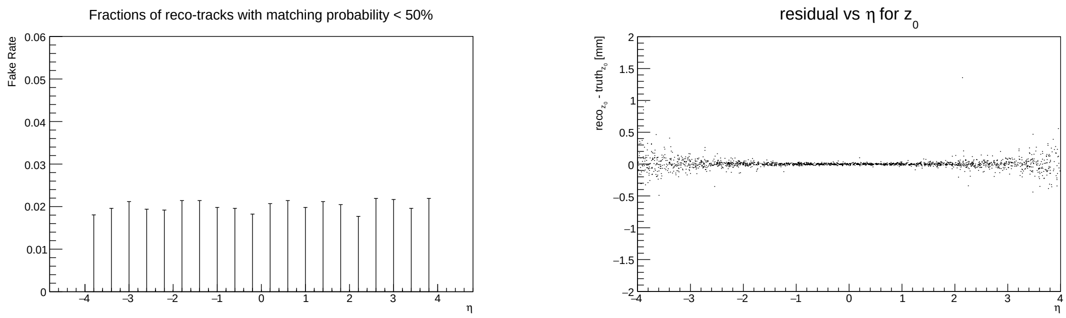

Figure 5: Fake rate (left) and residual of the measured values of a particle's impact parameter z_0 (right) as a function of η.

The performance of charged track reconstruction can be quantified by:

- The track reconstruction efficiency, defined as the ratio of the number of signal truth particles matched to reconstructed tracks divided by the number of signal truth particles. An ideal tracking algorithm would achieve perfect efficiency, i.e. reconstruct all particles in the sample.

- The fake rate, which is the fraction of fake hits combinations over the total reconstructed tracks. A collection of hits that is not produced by a single particle but returned by the tracking algorithm is referred to as fake. Usually, if a track contains 50% and more of hits from different sources, then it is labelled as fake.

High efficiencies and low fake rates are observed across a wide range of pseudorapidities, as shown in Figures 4 and 5. The residuals, which represent the small differences between measured and expected values, being close to zero, indicate that the FTF is successful in making accurate track reconstructions. This aligns with what we would expect from a well-optimized tracking algorithm. However, it's worth noting that the residuals tend to be slightly larger for higher η values, possibly due to material effects.

3.2 Single Muon Sample with 100 GeV

Events containing single muons with constant transverse momentum of 100 GeV have been considered. All tracks are required to be within the fiducial region defined by $p_{\mathrm{T}}^{track} > 0.5$ GeV and $\eta < 4$. The required number of hits per track is *at least* 7 for the entire range of $\eta < 4$.

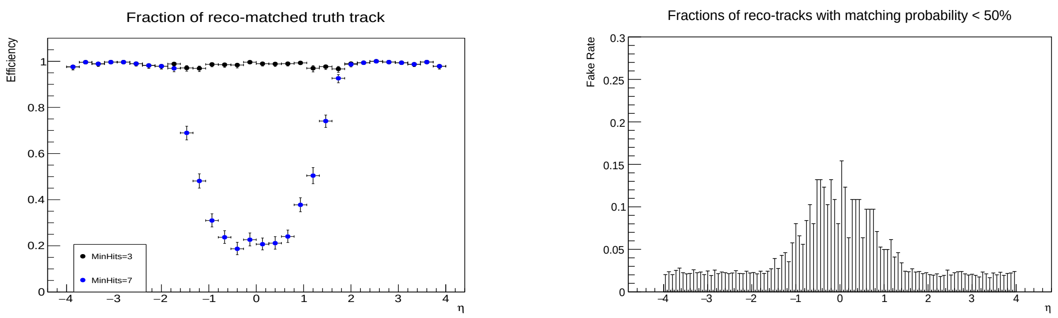

Figure 6: Track reconstruction efficiency, for different values of *MinHits*, as function of η (left). Fake rate versus η for *MinHits*=7 (right).

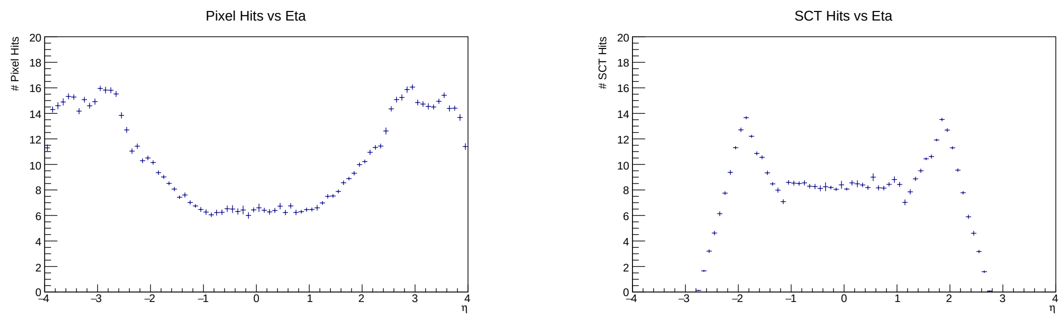

Figure 7: Average number of hits in silicon pixel layers and strip detector of the ITk, per reconstructed track, as function of η.

By keeping fixed the minimum p_{T}^{track}, *MinHits* (mininum number of hits that a particle track must have in order to be considered successfully reconstructed), were increased. The observed decreasing trend can be explained as follows: not all the particles may generate enough hits to satisfy a stronger requirement. This translates into a higher rejection rate, which means reduction of successfully reconstructed tracks, so into a decrease in efficiency. But in this case, if we increase *MinHits* from 3 to 7, we observe in Figure 6 that the efficiency decreases by approximately 70%. This indicates that FTF creates a set of three hits (a triplet), but it struggles to generate more complex tracks with high efficiency. It's worth noting that even though the efficiency decreases significantly, the fake rate remains low. Moreover the reconstructed tracks have an almost identical amount of hits in the pixel and strip detector as shown in Figure 7. The sample under study consists of highly energetic muons that are less likely to scatter or interact significantly with the detector material. These results show algorithm's limitations in handling highly energetic muons that, despite their high energy, often leave more than three traces in the detector. Further investigation and optimization of the fast track finder algorithm is required, especially when dealing with such energetic particles.

3.3 Single lepton events from $t\bar{t}$ pairs productions

Top quark pairs can be produced through the annihilation of a quark and an anti - quark ($q\bar{q}$ *annihilation*) or through the fusion of two gluons (ggF). The SM the top quark decays almost 100% of the time into a b quark and a W boson ($t \to bW$). The W boson will subsequently decay to either two quarks ($W \to q\bar{q}$) which are observed as jets of particles, or to a charged lepton and a neutrino ($W \to l\nu$). Three final state topologies are defined for $t\bar{t}$ events:

1. Dilepton:

$$t\bar{t} \to bW^+\bar{b}W^- \to bl^+\nu\bar{b}l^-\bar{\nu} \tag{1}$$

2. Lepton+jets:

$$t\bar{t} \to bW^+\bar{b}W^- \to bq\bar{q}'\bar{b}l^-\bar{\nu} \tag{2}$$

$$t\bar{t} \to bW^+\bar{b}W^- \to l^+\nu bq\bar{q}'\bar{b} \tag{3}$$

3. All jets:

$$t\bar{t} \to bW^+\bar{b}W^- \to bq\bar{q}'\bar{b}q_1\bar{q}_2 \tag{4}$$

All tracks are required to be within the fiducial region defined by $p_{\mathrm{T}}^{track} > 0.5$ GeV and $|\eta| < 4$. The required number of hits per track is *at least* 7 for the entire range of $|\eta| < 4$. In Figure 8 all events in which we have just one lepton among the final decay products are considered.

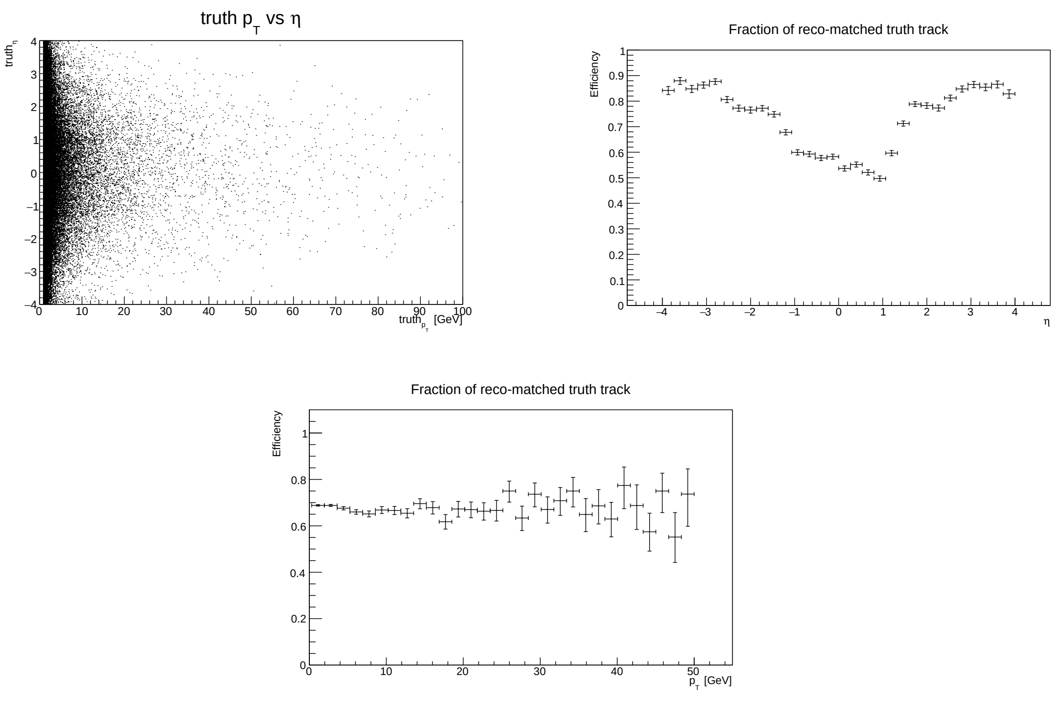

Figure 8: Distributions of truth p_{T} versus η, track reconstruction efficiency versus η and versus p_{T} are shown. All particles in the final state are included in the analysis.

The decrease in efficiency, especially in the central regions, and the higher efficiency for high η values can be attributed to the effect of high pile-up $\langle\mu\rangle = 200$. In the central regions of the detector the track density is higher, there may be challenges in reconstructing tracks due to interference with tracks from other events.
On the other hand this distributions show no significant fake rate, as we can see from Figure 9.

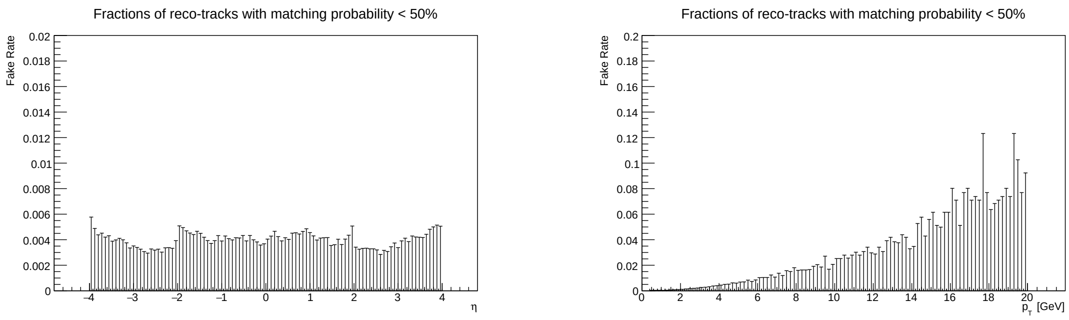

Figure 9: Fake rate versus η (left) and versus p_T (right).

The samples were then filtered to select events containing single muons and algorithm was setted to reconstruct only tracks of these muons. Obtained results are presented in Figure 10. This restriction leads to a more limited sample of events compared to the full set of single lepton events, resulting in a reduced number of events available for analysis. For this reason, associated statistical uncertainties are larger compared to those obtained with the complete dataset.

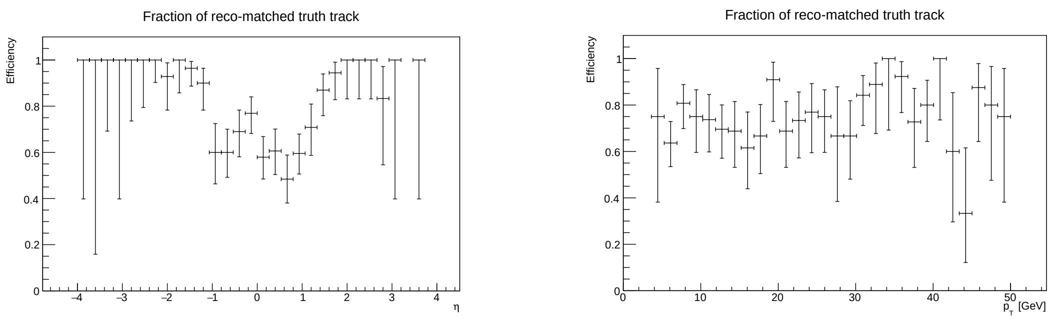

Figure 10: Track reconstruction efficiency of muons versus η (left) and versus p_{T} (right).

4 Conclusions

The performance of FTF algorithm with ITk geometry has been presented, focusing mainly on muons. Limitations of the algorithm were observed in handling complex tracks when dealing with highly energetic particles and with high values of pile up. Despite a significant decrease in efficiency, the algorithm maintained low fake rates. Future work should focus on optimization strategies to address these limitations and improve performance.

5 The future of HLT computing: exploiting GPUs

ATLAS is putting effort to include multithread computation devices in the Trigger and Data AcQuisition (TDAQ) system. GPUs (Graphics Processing Units) are one of the technologies under investigation.
For instance, when examining RoIs, certain characteristics stand out, such as a *high data locality* and a *high arithmetic density*, making them particularly well-suited for GPU-based parallelization. A redesign of track reconstruction algorithms is required to exploit this parallelization. While the potential is clearly there to speed up a specific algorithm by offloading it to a GPU, the effect on the performance of the overall system is not easy to determine.
Let's take a closer look at a specific example within the High-Level Trigger (HLT) system. In HLT the reconstruction of muon candidates consists of a three-step process:

- A track segment of a charged particle is reconstructed in the Muon Spectrometer (MS).
- Next, this track segment is matched to a charged particle track that was previously reconstructed in the Inner Detector.
- Lastly, evaluation of energy deposits in the electromagnetic and hadronic calorimeters occurs.

Now, if we focus for example on the third step and consider the possibility of running this algorithm on a GPU, the execution process can be summarized as follows:

1. The algorithm starts by accessing information from the calorimeter cells.
2. Next, it formats the necessary information into a memory buffer.
3. The buffer needs to be transferred to the server (APE), which then sends it to the GPU.
4. The algorithm is executed on the GPU using the CUDA programming language.
5. Finally, the results of the algorithm are transferred back into the original framework through APE.

A critical point in this process is establishing effective CPU-GPU communication. In the case of algorithm execution on a CPU, we directly move from step 1 to step 4, omitting steps 2 and 3, which are specific to data transfer on the GPU. What we have to compare is the trigger algorithm execution latency with respect to standard execution on CPU. This will imply the translation into a parallel computing programming language (CUDA) [5] of the original algorithm and the optimization of the different tasks that can be naturally parallelized, which is currently underway. Moreover, ongoing studies concern the integration of machine learning algorithms into the HLT workflow, with a focus on AI technology available from industry. This integration aims to improve pattern recognition, tracking, and other tasks crucial for particle physics analysis.

References

[1] ATLAS Collaboration, “High-Luminosity Large Hadron Collider (HL-LHC): Technical design report”, 2020 CERN

[2] ATLAS Collaboration, “Technical Design Report for the ATLAS Inner Tracker Pixel Detector”, 2017 CERN

[3] Fabian Klimpel, "Fast tracking for the HL-LHC ATLAS detector", ATL-PHYS-PROC-2020-044, 2020. arXiv:2004.13447 [hep-ex]

[4] ATLAS Collaboration, "The ATLAS Inner Detector trigger performance in pp collisions at 13 TeV during LHC Run 2," Eur. Phys. J. C 81, 156 (2021) https://link.springer.com/content/pdf/10.1140/epjc/s10052-021-09920-0.pdf.

[5] NVIDIA, Vingelmann, P. Fitzek, F.H.P., 2020. CUDA, release: 10.2. 89, Available at: https://developer.nvidia.com/cuda-toolkit