The performance and development of the ATLAS Inner Detector Trigger

A. Washbrook

SUPA, School of Physics and Astronomy, The University of Edinburgh,
James Clerk Maxwell Building, Mayfield Road, Edinburgh, EH9 3JZ, United Kingdom
E-mail: awashbro@ph.ed.ac.uk

ABSTRACT: A description of the ATLAS Inner Detector (ID) software trigger algorithms and the performance of the ID trigger for LHC Run 1 are presented, as well as prospects for a redesign of the tracking algorithms in Run 2. The ID trigger HLT algorithms are essential for a large number of signatures within the ATLAS trigger. During the shutdown, modifications are being made to the LHC machine, to increase both the beam energy and luminosity. This in turn poses significant challenges for the trigger algorithms both in terms of execution time and physics performance. To meet these challenges the ATLAS HLT software is being restructured to run as a single stage rather than in the two distinct levels present during the Run 1 operation. This is allowing the tracking algorithms to be redesigned to make optimal use of the CPU resources available and to integrate new detector systems being added to ATLAS for post-shutdown running. Expected future improvements in the timing and efficiencies of the Inner Detector triggers are also discussed. In addition, potential improvements in the algorithm performance resulting from the additional spacepoint information from the new Insertable B-Layer are presented.

KEYWORDS: Trigger algorithms; Software architectures (event data models, frameworks and databases); Online farms and online filtering; Computing (architecture, farms, GRID for recording, storage, archiving, and distribution of data)

1On behalf of the ATLAS collaboration
1 Introduction

The ATLAS detector [1] is one of two general purpose experiments at the LHC [2]. During the 2010-11 running period the LHC was operated with a \( pp \) collision energy of 7 TeV and increased to 8 TeV during 2012 running. Over this data-taking period (referred to as Run 1) a combined luminosity of 25 fb\(^{-1}\) was recorded by ATLAS. Since the first collisions in 2010 the instantaneous luminosity has increased by many orders of magnitude. This was achieved through an increase in the number of colliding bunches and by significantly increasing the number of protons in each bunch. The average number of distinct \( pp \) interactions per bunch crossing (referred to as \( <\mu> \)) during 2012 running was 21, with multiplicities higher than 35 being not uncommon.

During the current LHC Long Shutdown 1, the LHC machine and experiments are being upgraded in preparation for Run 2. The LHC machine will be upgraded to a collision energy of 13-14 TeV, together with an improvement in the instantaneous luminosity, which will lead to an increase in the average number of interactions per bunch crossing to \( <\mu> = 50 \) and over. The increase in collision energy and instantaneous luminosity necessitate significant improvements to the ATLAS detector, and particularly the trigger and data acquisition system.

1.1 The ATLAS Inner Detector

The ATLAS Inner Detector (ID) provides precise tracking and vertexing close to the point of interaction and enables accurate identification and measurement of objects such as electrons, muons, tau leptons and heavy flavour jets.

The ID is formed of three sub-detectors arranged in concentric layers with each using different tracking technologies. The Pixel detector is located closest to the beam pipe and consists of 1744 silicon pixel modules, each containing 46,080 50 \( \times \) 400\( \mu \)m pixels. Modules are arranged into three concentric layers of silicon pixel sensors in the barrel and the end-caps. The Semi-Conductor...
Tracker (SCT) is an additional silicon detector surrounding the Pixel detector which consists of 4088 microstrip detectors each containing 780 readout strips of 80 µm pitch. These are arranged in back-to-back wafers with a 40 mrad stereo angle in four layers in the barrel and nine in the end-caps. At the outermost layer of the ID is the Transition Radiation Tracker (TRT). This is a cylindrical detector consisting of 320,000 4 mm diameter straw drift tubes arranged parallel to the beam-pipe in the barrel and radially in the end-caps.

1.2 The ATLAS Trigger

During Run 1 the ATLAS Trigger [3] consisted of three distinct, sequential trigger levels - the hardware Level 1 (L1) trigger, running on limited granularity data predominantly from the muon spectrometer and calorimeter subsystems, and the two high level trigger (HLT) systems - the Level 2 (L2) and the Event Filter (EF), each running on its own large farm of commodity processors. The trigger system must reduce the rate of events from the nominal 40 MHz bunch crossing rate to a value suitable for recording online.

When an event is accepted by the L1 trigger, data are read out from the pipelines on the detector front end electronics into custom ReadOut Buffers (ROBs), where they are stored for access by the L2 processors. To reduce the rate at which data must be read out from the ROBs, Regions of Interest (RoIs) are identified by the L1 trigger which contain features of interest which merit further processing. The RoI based data access reduces the data that must be read out to approximately 2% of the full detector volume.

An L2 decision, if the event is to be kept, is followed by the EF reconstruction which runs modified versions of the offline reconstruction algorithms in the RoIs. Between 2010-2012 the EF output rate was in range of 350 to 1000 Hz.

2 Tracking performance

The efficiency of the HLT for electrons was measured using the tag and probe method. In this study, a tag lepton was required to be reconstructed in the electromagnetic calorimeter and the ID, and a probe lepton was required to be reconstructed in the calorimeter only (although leptons with ID hits are not vetoed). These were then matched to offline electron tracks in order to reduce the contribution from background processes. No tracking selection cuts were applied to the probe electron in the trigger to avoid bias.

The efficiency of a given trigger algorithm depended on whether there was a match between the track identified by the ID trigger and the offline track matched to the probe which satisfied the requirement of \( \Delta R = \sqrt{\Delta \phi^2 + \Delta \eta^2} < 0.03 \). In addition, the tag and the probe lepton were required to have \( p_T > 24 \text{ GeV} \) and \( 15 \text{ GeV} \) respectively, \( |\eta| < 2.5 \), and an invariant mass of the pair between 70 and 120 GeV to identify genuine electrons from the decay of a \( Z \) boson.

Figure 1 shows the efficiencies resulting from this study on a representative sample of the 2012 data. The efficiency is shown separately for the L2 and EF ID trigger tracking (red and black respectively). It can be seen that a high efficiency is achieved at high \( p_T \) (figure 1(a)) for both the L2 and EF despite the challenging conditions in this data-taking period. Figure 1(b) shows the efficiency as a function of the ratio of track \( p_T \) to calorimeter \( E_T \). Since the calorimeter is sensitive to radiation from electrons photons, this ratio is a useful measure of the amount of bremsstrahlung.
an electron has undergone. Bremsstrahlung represents a challenge for tracking since it leads to changes in track curvature which must be incorporated into the track fitting model. However, even for electrons losing 50% of their momentum in bremsstrahlung, the trigger tracking is over 98% efficient at L2, and over 99% efficient at the EF stage. In addition, the performance as a function of the average number of interactions per bunch crossing ($<\mu>$) reflected improvements made in the short shutdown before 2012, with very little dependence of efficiency on $|\eta|$ and an excellent efficiency seen for high pile-up data.

3 IBL

The Insertable B-Layer (IBL) [4] is a new Pixel detector to be installed during the 2013-14 LHC shutdown (LS1), and will become the new innermost ID barrel layer. The current innermost layer of the Pixel detector is 50.5 mm from the nominal interaction point (IP) at the centre of the detector, whilst the IBL will be situated 25.7 mm from the IP. The IBL improves vertexing performance and impact parameter resolution, and adds robustness against missed hits and disabled modules in track identification; all of which are extremely beneficial in high pileup conditions.

Whilst the HLT software tracking algorithms will also be updated and restructured during this period, it was instructive to measure the performance of the L2 and EF triggers as they were during Run 1 operations with the IBL included in the data simulation. By including a possible additional hit from the IBL for each track, close to the beam line, it was expected that the reconstruction efficiency should increase and the resolution on the impact parameter measurement should improve.

For this study 14 TeV samples of $t\bar{t}$ events generated using MC@NLO [5] were used which allowed the tracking to be evaluated under reasonably high multiplicity conditions. The samples were filtered to select events containing single muon and di-muon final states and tracks reconstructed by the triggers were required to be in the range $|\eta| < 2.5$, and to have at least one hit in the silicon detectors. Resolutions for the transverse impact parameter, $d_0$, as a function of signed $p_T$ for the L2 and EF algorithms with and without inclusion of hits from the IBL are shown in fig-
Figure 2. L2 reconstructed $d_0$ resolution as a function of (a) signed $p_T$ and (b) $\eta$. The hollow (red) points are without IBL, the solid points are with IBL. From Reference [6].

Figure 3. EF reconstructed $d_0$ resolution as a function of (a) signed $p_T$ and (b) $\eta$. The hollow (red) points are without IBL, the solid points are with IBL. From Reference [6].

An improvement of around 25% is seen and a similar improvement is also observed for other track parameter resolutions [6].

4 Software development and optimisation

A schematic of the planned redesign of the HLT software is shown in figure 4. For Run 2 the two distinct HLT stages (L2 and EF) will be modified to run on a single node on the HLT computing farm. This will reduce the overall data volume that needs to be requested by the HLT system, since data requested by the L2 algorithms will no longer need to be requested again when building the event for the EF processing. The operation of the HLT will still be similar to the separate tiers used before, with a fast tracking stage followed by more detailed tracking, but with increased sharing of information between the two stages. This single node operation will also provide an opportunity to redesign the tracking algorithms to combine the reconstruction currently performed separately at L2 and the EF in a more optimal way. There will also be input to the HLT decisions by new detector
subsystems such as the IBL and Fast Tracker (FTK) [7]. The FTK is a new hardware based track finder which will be installed during Run 2 operations and perform tracking in the ID for events after acceptance by L1 but before L2 execution.

4.1 Optimisation studies

The timing performance of the ID Trigger software was evaluated using profiling tools to identify possible areas where code optimisation would be beneficial. A selection of HLT software benchmark test jobs were created to process a representative sample of input data collected from the 2012 running period to determine where possible performance improvements could be made. These test jobs were configured to reconstruct tracks in the full volume of the ID (using a single RoI covering the whole detector) so that the impact of performance hotspots could be highlighted more prominently. A dedicated testbed was configured to run HLT jobs exclusively, to ensure that no other active applications running concurrently were able to affect the timing results.

Accurate timing information was gathered from the CPU monitoring and timing service routines available in the ATLAS software framework. The relative times spent per event for the algorithms from one of the track reconstruction strategies used at Level 2 is shown in figure 5. Note that absolute timing values depend on the underlying hardware; thus those from the dedicated testbed are slightly different from those observed on the online systems.

Test jobs were then profiled using the Callgrind [8] tool which collected the number of instruction fetches (used as a measure of cost) and the total number of calls for each function used in the execution of the code. Despite the large amount of functions profiled (over 2000 functions) it was found that only a small fraction of functions contributed to the majority of instruction fetches. Functions with the highest number of CPU instruction fetches per event are shown in figure 6.

In addition to call-level profiling data, additional profiling tools were used to determine whether the CPU time spent by each algorithm was being used efficiently. The Linux performance counter subsystem (perf) and the Generic Optimization Data Analyzer (GOoDA) [9] package enabled the sampling of CPU hardware counters during the execution lifetime of test jobs. GOoDA provided an analysis of the low-level counter information collected by perf to allow inefficiencies in workloads, such as cache-misses and branch mis-prediction, to be identified and attributed to specific sections of source code.
Figure 5. Relative time per event spent in Level 2 Strategy A algorithms. The timing values were taken from the processing of 1367 events collected in the 2012 running period. From Reference [6].

Figure 6. Number of CPU instruction fetches per event generated collected by the callgrind profiling tool. Functions with the highest number of instruction fetches are illustrated. The software library containing each function is shown in parentheses. From Reference [6].

Figure 7 illustrates the CPU cycle data collected for one of the costly functions profiled in the ID Trigger code. In this case branch mis-prediction is the cause of 21% of the total unhalted cycles measured executing this function and was therefore considered as an area for potential optimisation. An isolated copy of the Z-finder algorithm was modified with the aim of reducing the amount of stalled CPU cycles due to branch mis-prediction. A significant improvement in the timing per event for this test code can be seen in figure 8 where the mean execution time per RoI is 2.1 ms with the unmodified code and improves to 1.1 ms after optimisation.

5 Outlook

The high number of interactions and resulting large track multiplicity in each event expected in future data-taking at the LHC provides a very challenging environment for tracking in the ATLAS trigger. The opportunity afforded by the 2013-14 long shutdown to redesign and upgrade the AT-
To gain significant performance improvements from vectorization it was therefore necessary to man-
ual vectorization to reduce branch mis-prediction.

Figure 7. Total number of unhalted CPU cycles sampled during 100 processed events in the Z-Finder
ID Trigger algorithm. Stalled cycles are a subset of the total unhalted cycles and load latency, branch-
misprediction and instruction latency are a subset of the total stalled cycles. From Reference [6].

Figure 8. Comparison of execution time for two versions of the Z-Finder algorithm, before and after opti-
misation to reduce the measured CPU branch mis-prediction rate. From Reference [6].

LAS ID trigger is being exploited with progress in a complete redesign of the tracking strategies
being made. In particular, the ability to run the HLT on a single CPU node is enabling a more
staged, “pipelined” approach to the reconstruction. Incorporating modifications to make use of the
additional spacepoint information from the IBL will result in significantly better resolution for the
transverse impact parameter and improved performance. In addition, optimisation studies continue
to be made with existing ID Trigger algorithms and are being extended to the post-shutdown algo-
rithms. Taken together, these changes will allow the ATLAS ID trigger to continue to perform as
well, or better than in Run 1 despite the significantly higher luminosities expected when the LHC
restarts after the shutdown.
References

[1] ATLAS collaboration, The ATLAS Experiment at the CERN Large Hadron Collider, 2008 JINST 3 S08003.

[2] L. Evans and P. Byant, LHC Machine, 2008 JINST 3 S08001.

[3] ATLAS collaboration, Performance of the ATLAS Trigger System in 2010, Eur. Phys. J. C 72 (2012) 1849 [arXiv:1110.1530].

[4] M. Capeans et al., ATLAS Insertable B-Layer Technical Design Report, CERN-LHCC-2010-013, ATLAS-TDR-19, CERN, Geneva (2010).

[5] S. Frixione and B.R. Webber, The MC@NLO 3.2 event generator, hep-ph/0601192.

[6] ATLAS collaboration, Studies for the development of the Inner Detector trigger algorithms at ATLAS, ATL-DAQ-PUB-2013-002, CERN, Geneva (2013).

[7] M. Shochet, L. Tompkins, V. Cavaliere, P. Giannetti, A. Annovi and G. Volpi, Fast TracKer (FTK) Technical Design Report, CERN-LHCC-2013-007. ATLAS-TDR-021, CERN, Geneva (2013).

[8] N. Nethercote and J. Seward, Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation, Proceedings of ACM SIGPLAN 2007 Conference on Programming Language Design and Implementation (PLDI 2007), (2007).

[9] P. Calafiura, S. Eranian, D. Levinthal, S. Kama and R.A. Vitillo, GOoDA: The Generic Optimization Data Analyzer, J. Phys. Conf. Ser. 396 (2012) 052072.