

# Performance and development plans for the Inner Detector trigger algorithms at ATLAS

**Stewart Martin-Haugh**

Rutherford Appleton Laboratory Harwell Oxford, Didcot, OX11 0QX

E-mail: [stewart.martin-haugh@cern.ch](mailto:stewart.martin-haugh@cern.ch)

**Abstract.** A description of the design and performance of the newly re-implemented tracking algorithms for the ATLAS trigger for LHC Run 2, to commence in spring 2015, is presented. The ATLAS High Level Trigger (HLT) has been restructured to run as a more flexible single stage process, rather than the two separate Level 2 and Event Filter stages used during Run 1. To make optimal use of this new scenario, a new tracking strategy has been implemented for Run 2. This new strategy will use a FastTrackFinder algorithm to directly seed the subsequent Precision Tracking, and will result in improved track parameter resolution and significantly faster execution times than achieved during Run 1 and with better efficiency. The timings of the algorithms for electron and tau track triggers are presented. The profiling infrastructure, constructed to provide prompt feedback from the optimisation, is described, including the methods used to monitor the relative performance improvements as the code evolves. The online deployment and commissioning are also discussed.

## 1. Introduction

The Large Hadron Collider (LHC) [1] is, at the time of writing, being prepared for Run 2, a new period of data-taking. In Run 1 the LHC collided proton beams at two centre-of-mass energies: 7 and 8 TeV. In Run 2 this will be increased to 13 TeV. Additionally the instantaneous luminosity will increase from its peak Run 1 value of approximately  $8 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$  with a mean number of interactions per bunch crossings ( $\langle \mu \rangle$ ) equal to 21 to approximately  $2 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  with an average of  $\langle \mu \rangle = 46$ . The increase in energy and luminosity poses a particular challenge for the trigger system.

## 2. Inner Detector and Trigger Architecture

The ATLAS detector is one of two general-purpose detectors at the LHC and is described in more detail elsewhere [2]. It principally consists of an inner tracking detector, the Inner Detector (ID), electromagnetic and hadronic calorimeters and a muon spectrometer, as well as solenoidal and toroidal magnets. The ID plays a key role in the identification and measurement of objects, including electrons, muons, tau leptons and heavy flavour jets. The ID consists of three subdetectors: two silicon detectors (the Pixel and Semiconductor Tracker (SCT) detectors) and the Transition Radiation Tracker (TRT). The Pixel detector consists of an innermost layer, the Insertable B-Layer (IBL) situated 25.7 mm from the beamline, followed by three concentric layers of silicon pixel sensors, arranged axially (radially) in the barrel (endcap). The IBL is a new addition for Run 2. The SCT barrel consists of four concentric layers of silicon microstrips, while



the endcap consists of nine layers of silicon microstrips. The Pixel and SCT provide tracking over the pseudorapidity range  $|\eta| < 2.5$ . The Transition Radiation Tracker (TRT) is a cylindrical detector extending to  $|\eta| < 2.0$  consisting of 320,000 straw tubes filled with a  $\text{XeCO}_2\text{O}_2$  gas mixture, around a central tungsten wire. A typical TRT track will have approximately 36 hits, which allows improved estimation of track parameters when combined with the Pixel and SCT hits.

### 2.1. ATLAS Trigger architecture

During Run 1, ATLAS used a three-level trigger system [3]: an initial Level 1 hardware stage (L1) followed by the High Level Trigger (HLT) consisting of two software stages, Level 2 (L2), and the Event Filter (EF). The parameters of the different trigger levels are listed in Table 1.

Trigger level	Level 1	Level 2	Event Filter
Input rate	20 MHz	70 kHz	5-6 kHz
Output rate	70 kHz	5-6 kHz	700 Hz
Decision time	$< 2.5 \mu\text{s}$	75 $\mu\text{s}$	1 s

**Table 1.** Parameters of the three trigger levels during Run 1.

The trigger system has undergone several major upgrades in preparation for Run 2. One of the most significant is the merging of the two L2 and EF of the HLT into a single stage running on a single combined HLT farm. This has important consequences for trigger algorithm design in Run 2 - in particular, data preparation was performed separately for L2 and EF, while it is now a single process for the HLT. The design parameters of the Run 2 trigger system are shown in Table 2.

Trigger level	Level 1	HLT
Input rate	40 MHz	100 kHz
Output rate	100 kHz	1 kHz
Decision time	$< 2.5 \mu\text{s}$	200 ms

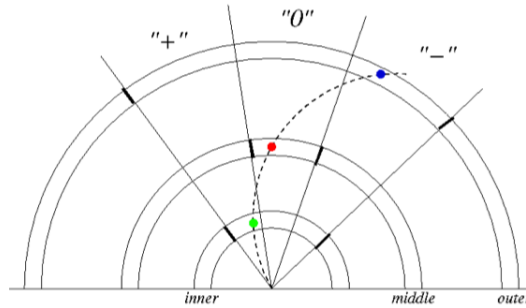
**Table 2.** Parameters of the three trigger levels during Run 2.

## 3. Inner Detector Trigger for Run 2

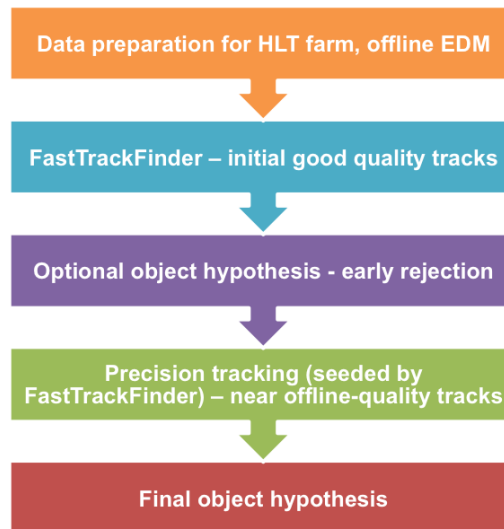
In Run 1 several different algorithms [4] were used in the L2 trigger. For Run 2 a single L2-like algorithm, known as *FastTrackFinder* (FTF) has been developed in order to quickly provide medium-quality tracks which are then used to seed a *Precision Tracking* (PT) stage, which is a modified version of the EF tracking.

During the pattern recognition stage, a search for triplets of spacepoints (*track seeds*) is performed in bins of  $r$  and  $\phi$ , as shown in Figure 1. The track parameters at the perigee ( $d_0$ ,  $z_0$ ,  $\phi_0$ ,  $\eta$  and  $p_T$ ) are estimated, and the parameters in the transverse plane ( $d_0$ ,  $\phi$ ,  $p_T$ ) are transformed by a conformal mapping [5] to improve numerical stability. The track seeds are then passed to an offline track-finding algorithm with settings optimised for speed. A simple algorithm to remove duplicate tracks is applied, selecting tracks of higher quality. The preliminary tracks are then given to a fast Kalman filter-based track fitter [6]. For speed, TRT hits are ignored in the FTF. The PT stage takes the input FTF tracks, applies a more detailed algorithm for duplicate track candidates, tries to extend the track into the TRT, and finally refits the track using a more precise global  $\chi^2$  fitter [7]. The Run 2 strategy is shown schematically in Figure 2.

During Run 1, because the EF processing was running on a separate CPU to the L1 tracking, the EF tracking needed to perform the pattern recognition stage anew. The Run 2 strategy does not require the second data preparation step and offers more flexibility by not requiring an initial object hypothesis. Additionally, the Run 2 strategy only performs track finding starting from spacepoints a single time. Overall the Run 2 strategy offers improved timing and flexibility without sacrificing performance.

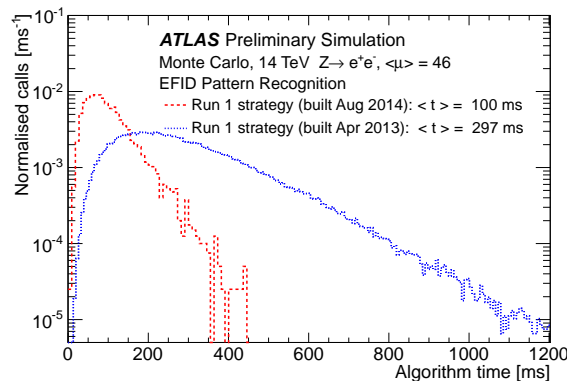


**Figure 1.** Illustration of an aspect of the FastTrackFinder track seeding - spacepoints in neighbouring  $\phi$  bins above and below a middle spacepoint are combined into a triplet of spacepoints [8].



**Figure 2.** Schematic of the Run 2 ID trigger strategy [8].

In addition to the constraints to the trigger system in Run 2, there are strong constraints on the time taken to perform offline reconstruction. For this reason an extensive programme of software optimisation was undertaken between Run 1 and Run 2. This provides tangible benefits for the ID trigger since much of the ID trigger software is shared with offline ID reconstruction. Some of the improvement in execution time arise from a switch to a newer compiler (GCC 4.3 to 4.8), some from the change to 64-bit architecture, and some from the replacement of the CLHEP [9] linear algebra library by Eigen [10]. Hotspots in the code were identified by profiling and then optimised. Figure 3 shows the difference in timing for a Run 1 ID trigger strategy when executed in a Run 1 software release (in blue) and a Run 2 software release (in red), showing clear improvements in execution time from updates to offline software.



**Figure 3.** CPU timing for the ID trigger run 1 strategy, showing the effect of CPU optimisation between the Run 1 and Run 2 software releases, shown in blue and red respectively [11].

### 3.1. Two-step tracking

The CPU timing of track seeding and finding algorithms generally exhibits a non-linear dependence on the number of spacepoints that must be processed. In the ATLAS HLT this effect is mitigated by limiting tracking to geometric Regions of Interest (RoIs), which are defined by boundaries  $\eta$ ,  $\phi$  and  $z$  – the  $z$ -coordinate along the beamline. The RoIs are seeded by the L1 calorimeter and muon triggers.

In Run 2, HLT strategies can use additional, more precise RoIs in order to mitigate the CPU cost of track finding. This approach is known as *two-step tracking*, and has been implemented in particular in the trigger for hadronic tau lepton decays, where tracking in single large RoI has a large CPU cost. The two-step tracking sequence for tau decays is as follows:

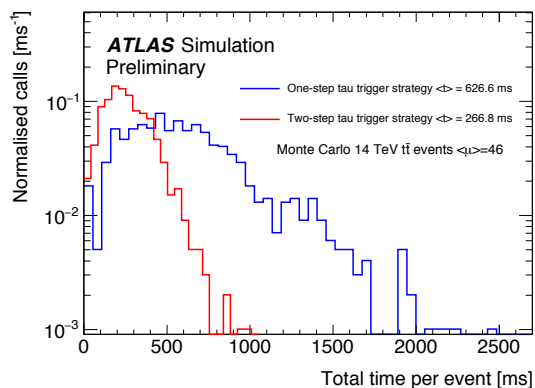
- (i) Reject events without a high- $p_T$  lead track in the RoI with  $\Delta\eta \times \Delta\phi \times \Delta z = 0.1 \times 0.1 \times 225$  with respect to the central RoI coordinates
- (ii) Find additional tracks in  $\Delta\eta \times \Delta\phi = 0.2 \times 0.2$  RoI within  $\Delta z = 10$  mm of lead track

The first RoI is constrained more tightly in  $\eta$  and  $\phi$ , and very loosely in  $z$ , while the second RoI is constrained in  $\Delta z$ . In the one-step tracking case, a single RoI of size  $\Delta\eta \times \Delta\phi \times \Delta z = 0.4 \times 0.4 \times 225$  mm is used. Figure 4 shows the CPU timing of one- and two-step trigger strategies measured with an inclusive  $t\bar{t}$  sample and  $\langle\mu\rangle = 46$ , comparable to the expected Run 2 luminosity. The mean timing is 626.6 ms for the one-step tracking strategy and 266.8 ms for the two-step tracking strategy. Figure 5 shows the mean CPU timing for  $t\bar{t}$  samples with 46, 69 and 138 additional interactions. The one-step tracking strategy is seen to scale non-linearly, while the two-step tracking strategy scales linearly with additional interactions over this range. These studies with higher  $\langle\mu\rangle$  will be relevant for upgrades of the ATLAS trigger for the harsher running conditions of the proposed High-Lumi LHC.

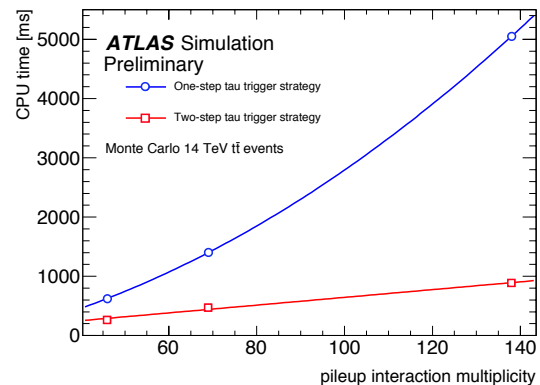
### 3.2. Profiling and validation

To prepare the rewritten ID trigger code for Run 2, an extensive programme of profiling and validation has been undertaken. The different stages used are as follows:

- Static code analysis using Coverity [12]
- Dynamic code analysis using Valgrind, AddressSanitizer[13] and UndefinedBehaviorSanitizer [14]
- Regular CPU timing measurements on dedicated machines similar to the HLT farm nodes
- Profile-guided optimisation using Callgrind [15] and GooDA [16]



**Figure 4.** Distribution of CPU timing for one- and two-step tracking in blue and red respectively [11].



**Figure 5.** Average CPU timing of one- and two-step tracking in three pile-up scenarios:  $\langle \mu \rangle = 46, 69$  and  $138$  [11]. In this regime the two-step tracking shows a linear dependence on pile-up.

- Reconstruction of cosmic ray events
- Performance measurements on Run 1 data and simulated events

This programme allows the correctness, physics performance and CPU processing time of the ID trigger algorithms to be established in preparation for Run 2.

### 3.3. Conclusions

The ID trigger has been extensively upgraded in preparation for Run 2, and significant performance improvements have been achieved. The performance improvements come from general software improvements, the new FTF and Precision Tracking algorithms and more flexible algorithm strategies, particularly the new two-step tracking strategy. The ID trigger is ready to meet the challenges of Run 2 data taking.

## References

- [1] Evans L and Bryant P 2008 *JINST* **3** S08001
- [2] ATLAS Collaboration 2008 *JINST* **3** S08003
- [3] ATLAS Collaboration 2012 *Eur.Phys.J.* **C72** 1849
- [4] Bernat P 2012 URL <https://cds.cern.ch/record/1458060>
- [5] Yepes P 1996 *Nucl. Inst. Meth. A* **380** 582 – 585
- [6] Frühwirth R 1987 *Nucl. Inst. Meth. A* **262** 444–450
- [7] Cornelissen T G, Elsing M, Gavrilenko I, Laporte J, Liebig W *et al.* 2008 *J. Phys. Conf. Ser.* **119** 032013
- [8] Martin-haugh S 2015 URL <https://cds.cern.ch/record/2007272>
- [9] Lönnblad L 1994 *Computer Physics Communications* **84** 307 – 316 ISSN 0010-4655
- [10] Guennebaud G 2013 URL [http://downloads.tuxfamily.org/eigen/eigen\\_CGLibs.Giugno.Pisa.2013.pdf](http://downloads.tuxfamily.org/eigen/eigen_CGLibs.Giugno.Pisa.2013.pdf)
- [11] ATLAS Collaboration URL <https://twiki.cern.ch/twiki/bin/view/AtlasPublic/HLTTrackingPublicResults>
- [12] Bessey A, Block K, Chelf B, Chou A, Fulton B, Hallem S, Henri-Gros C, Kamsky A, McPeak S and Engler D 2010 *Communications of the ACM* **53** 66–75
- [13] Serebryany K, Bruening D, Potapenko A and Vyukov D 2012 AddressSanitizer: A Fast Address Sanity Checker *USENIX Annual Technical Conference* pp 309–318
- [14] LLVM Collaboration URL <http://clang.llvm.org/docs/UsersManual.html>
- [15] Weidendorfer J 2008 Sequential performance analysis with callgrind and kcache-grind *Tools for High Performance Computing* (Springer) pp 93–113
- [16] Calafiura P, Eranian S, Levinthal D, Kama S and Vitillo R 2012 *J. Phys. Conf. Ser.* **396** 052072