Improvements to ATLAS Inner Detector Track reconstruction for LHC Run-3

Zachary Michael Schillaci[1], on behalf of the ATLAS Collaboration[*]

[1]Brandeis University

Abstract. This document summarises the main changes to the ATLAS experiment's Inner Detector Track reconstruction software chain in preparation of LHC Run 3 (2022-2024). The work was carried out to ensure that the expected high-activity collisions with on average 50 simultaneous proton-proton interactions per bunch crossing (pile-up) can be reconstructed promptly using the available computing resources. Performance figures in terms of CPU consumption for the key components of the reconstruction algorithm chain and their dependence on the pile-up are shown. For the design pile-up value of 60 the updated track reconstruction is a factor of 2 faster than the previous version.

1 Introduction

The reconstruction of charged particle trajectories (tracking) in the Inner Detector (ID) is a complex part of the ATLAS [1–3] experiment's event reconstruction chain, making it the most resource intensive component during Run 2 of the LHC (2015-2018). The ID is the closest detector component to the interaction point (IP) and consists of a high granularity pixel detector, a semiconductor tracking detector (SCT) as well as a transition radiation tracker (TRT). Given its close proximity to the interaction point (IP) and high granularity, the ID records up to 1500 hits per single proton-proton (p-p) collision, while each bunch crossing results in a number of simultaneous proton-proton collisions taking place (pile-up, $\langle\mu\rangle$). The average $\langle\mu\rangle$ during Run 2 ranged from 20 to 40, with a peak luminosity of $1.9 \times 10^{34}\text{cm}^{-2}\text{s}^{-1}$ [4], twice the original LHC design value. As a result, in an average bunch crossing event, about 30000 to 60000 hits need to be processed, decoded, and combined into clusters. The clusters then need to be combined into short track seeds that are subsequently attempted to be extended through the entire ID to identify the charged particles (tracks) and precisely reconstruct their trajectories. This represents a complex combinatorial problem, which increases in difficulty with pile-up. In addition to the computational effort, the quality of the reconstructed track candidates becomes challenging to maintain under high pile-up, as the high density of clusters leads to incorrect cluster-to-track association, potentially pulling the reconstructed trajectories away from their true values. Additionally, largely random collections of clusters can be reconstructed as tracks, which happens more frequently as the number of available clusters increases with pile-up. The timing required for tracking scales rapidly with $\langle\mu\rangle$. For the LHC Run 3 (2022-2024) data-taking, averages of around 50 interactions per bunch-crossing are expected, a significant increase compared to a mean value of 33.7 in Run 2. The ATLAS

[*]Copyright 2021 CERN for the benefit of the ATLAS Collaboration. CC-BY-4.0 license.

track reconstruction as operated during LHC Run 2 was optimised for $\langle\mu\rangle = 20$ and would hence be unfeasible to operate as-is under Run 3 conditions. Preparing the reconstruction to cope with such conditions was therefore a main focus of the ATLAS Run 3 reconstruction updates. One main direction not discussed here was the adoption of multi-threading to make more efficient use of the available resources. This however is not, a priori, expected to improve processing time required by the algorithms. In addition to this infrastructural change, a major effort [5] was carried out to improve the per-thread performance of track reconstruction while maintaining comparable or even superior quality of the reconstructed tracks. The changes made in this effort and their impact are described in this paper.

2 Track Reconstruction

ATLAS track reconstruction as performed during LHC Run 2 is extensively documented in other sources [7–11]. The software implementation was finalised in 2014 and tuned for the expected conditions during LHC Run 2. A brief overview is provided in the following.

The procedure starts with a pre-processing stage. Signals from adjacent channels in the Pixel and SCT subdetectors are combined into clusters which are interpreted as the deposits left by individual traversing charged particles. Pairs of one-dimensional SCT clusters on either side of a sensor module or individual pixel clusters are further converted into 3-dimensional space-points, with position uncertainties determined by the detector geometry and sensor pitch.

The primary ATLAS track reconstruction pass starts by forming so-called track seeds consisting of triplets of space-points in the Pixel or SCT sub-detectors which are compatible with originating from a charged particle track. Search roads (sets of detector modules that can be expected to contain clusters compatible with the seed) are built through the remaining detector based on the estimated seed trajectory, and the seeds are extended with additional clusters along the search road into silicon track candidates by means of a combinatorial Kalman Filter [12].

To resolve overlaps between track candidates and reject incorrect combinations of unrelated clusters ("fake tracks"), a dedicated ambiguity solution step is performed, which scores track candidates based on a range of quality criteria and rejects lower-quality candidates sharing a large number of associated hits with higher-quality ones. A limited number of shared hits is permitted to retain high performance in dense topologies such as cores of high-energy jets, where the separation between charged particles is expected to reach below the magnitude of the sensor pitch. The estimated cluster positions and their uncertainties are updated using a neural network based algorithm, and a probability is assigned for one, two, or at least three charged particles to have contributed to the cluster. Clusters judged to consist of more than one charged particle crossing are split among track candidates, with position and uncertainty estimates for each particle crossing provided by the algorithm.

The refined and purified track candidates resulting from the ambiguity resolution are then re-fit using a global χ^2 method to obtain the final, high-precision track parameter estimate. An extension of the track into the TRT subdetector is attempted, with a re-fit of the entire track being performed in case of a successful extension to profit from the additional measurements on track in particular for momentum resolution and particle identification.

This reconstruction pass is optimised for particles produced in the primary p-p interactions. To increase acceptance to particles produced at a greater distance to the beam line, such as electrons originating from photon conversions in the detector material, a secondary back-tracking pass is performed using the detector hits not already assigned to tracks from the primary pass. Here, track reconstruction is only attempted in regions of interest determined by deposits in the electromagnetic calorimeter. Unlike the first pass, this second pass starts with segments of hits in the TRT compatible with the region of interest. In presence of such a segment, short silicon track seeds consisting of two space-points are constructed in the Pixel and SCT subdetectors, and extended into track candidates using the same procedure as for the primary pass. A dedicated ambiguity resolution pass among the track candidates in the second pass and a re-fit of the resulting tracks including their TRT extension complete the second pass.

Further tracking passes are performed to reconstruct short tracklets from muons in $|\eta| > 2.5$, where only the pixel detector is traversed, as well as short tracks compatible with decaying, short-lived charged particles. In each case, only left-over hits from prior passes are used to limit combinatorial complexity.

After all track candidates have been reconstructed, the locations of the underlying *p*-*p* interactions (vertices) are identified by a dedicated vertex reconstruction procedure [13]. A first step obtains an initial position estimate for a vertex from the distributions of the z coordinate of the tracks' closest approach to the beamline. Then, a fit of the vertex location is performed taking into account all tracks loosely compatible with the initial position estimate. Before the changes reported in this paper, this was performed using an iterative procedure, constructing one vertex at a time and removing the associated tracks from consideration before repeating the procedure.

The track reconstruction procedure described above does not attempt to reconstruct tracks that have a very large distance of closest approach orthogonal to the beam line (transverse impact parameter, d_0). Measurements and searches requiring such tracks, for example to reconstruct decays of long-lived neutral particles within the Inner Detector volume, therefore run a dedicated version of track reconstruction [14] on a preselected sub-set of the collision data. This version uses the same algorithmic flow, but is configured with a wider search space in the transverse impact parameter, enabling reconstruction of displaced tracks at the price of drastically slowed execution speed.

3 Physics Performance and Software Optimisation

A number of changes to the tracking software were introduced in order to ensure that the computational performance and the size of the generated output will remain sustainable during LHC Run 3 data-taking. Apart from general algorithmic improvements, the guiding principle is to abort the track reconstruction as early as possible for candidates that are not expected to result in high-quality tracks, in order to minimise the number of executions of the downstream algorithms, thus saving time and resources.

A first step was to apply stricter requirements on track candidates when determining which ones to retain after the initial track-finding stage. Instead of seven silicon clusters on track, at least eight are required in Run 3, and the permitted transverse impact parameter range of silicon-seeded tracks is restricted to $|d_0| < 5$mm instead of $|d_0| < 10$mm. This reduces the acceptance of track reconstruction in terms of both displacement and production radius by a small fraction, but significantly reduces the number of low-quality tracks written to storage as well as the required number of iterations of the ambiguity resolution and TRT extension phases.

A large contribution of falsely reconstructed tracks was previously generated by the TRT-seeded back-tracking step. This was reduced by only performing the backtracking within regions of interest (ROI) seeded by energy deposits in the electromagnetic calorimeter ($E_T > 6$GeV), with the energy threshold chosen to be the largest value still sufficiently below the kinematic regime of electron reconstruction to avoid efficiency losses due to energy resolution effects. The recovery of late-appearing tracks from electron conversions, which is the main purpose of this reconstruction step, is hence only degraded at a negligible level since these topologies coincide with significant calorimetric deposits. However, the number of erroneously reconstructed track candidates is reduced and the execution speed of the backtracking phase is dramatically improved by a factor 20.

The seeding phase of the inside-out track reconstruction was optimised to prevent seeds unlikely to result in tracks from being passed into down-stream processing. This has a large impact on processing speed, since the number of executions of all following reconstruction steps is reduced. The optimisations include stricter requirements on the estimated impact parameters of the track seeds, narrower search roads used to extend the seeds, and a restriction of the number of mutually overlapping seeds to pass into further processing. The presence of a fourth pixel layer [2, 3] since 2015 is exploited by using confirmation space-points to detect promising seeds and treat them with preference. These changes have only a minor impact on the number of correctly reconstructed tracks, while significantly suppressing the occurrence of falsely reconstructed tracks and improving execution speed dramatically. The seeding strategy was further optimised by adapting the size of the angular regions within which seeds are formed to correspond to the track curvature resulting from the bending in the magnetic field of the detector expected to occur at the lowest track momentum to be reconstructed, instead of the wider angular regions used previously. This improves execution speed by reducing the number of combinatorial permutations to process during the seed finding stage, without significantly changing the number of tracks being reconstructed.

The iterative vertex finding algorithm described in Section 2 was replaced by an adaptive multi-vertex fitter algorithm [15], in which vertex candidates are allowed to compete for tracks in order to reduce the chance of nearby $p-p$ interactions being reconstructed as a single merged vertex. The initial vertex locations are estimated with high accuracy using a Gaussian resolution model for the track impact parameter. This updated algorithm is implemented within the ACTS [16] framework, and represents the first production use of this framework in an LHC experiment.

The TRT extension was sped up significantly by aborting the iterative track fit procedure early for candidates with insufficient compatible hits in the TRT. This change does not impact reconstruction efficiency or the rate of incorrectly reconstructed tracks, but speeds up the TRT extension step by nearly 30%.

Further execution speed was gained by carefully optimising the software implementation of each reconstruction step individually. Notable examples include a re-organisation of the search for holes on tracks performed as part of the precision fit, exploiting the navigation between detector surfaces already being performed by the track fit procedure, an optimisation of the space-point formation and the re-writing of parts of the Runge-Kutta propagator implementation used to extrapolate trajectories through the inhomogenous magnetic field of the detector to exploit vectorised instructions where possible.

The speed improvements achieved using the measures described above make it feasible to run an additional reconstruction pass to recover non-pointing tracks from displaced decays, using left-over hits left by the earlier passes, as part of the standard ATLAS track reconstruction. This removes the need for inefficient pre-selection and re-reconstruction to reconstruct these tracks in the Run 2 implementation of the software described earlier. In the following study of computational performance, the impact of this Large Radius Tracking (LRT) step will be pointed out separately, as it is not being run in the previous reconstruction the updated software is being compared to.

The reduction in the single-thread CPU timing per event for each of the optimisations listed above are shown in Figure 1 for a set of events recorded at the very high pile-up value of $\langle\mu\rangle = 90$. The reductions are given relative to the first iteration of the Run 3 reconstruction, which did not include any optimisations but was slower than the Run 2 implementation due to changes made to ensure the thread-safety of the code. Changes to the vertex finding, TRT extensions and further improvements are added under “Additional Optimisations”. The purple shaded area indicates the increase in the CPU per event by adding in the LRT tracking pass. For these challenging conditions, a factor 4 improvement in speed has been achieved. After the addition of LRT, near a factor of 3 reduction in the timing requirement per event is observed compared to the initial Run 3 software implementation before the changes discussed above, while the reconstruction efficiency is only affected in a marginal way as demonstrated in Section 5.

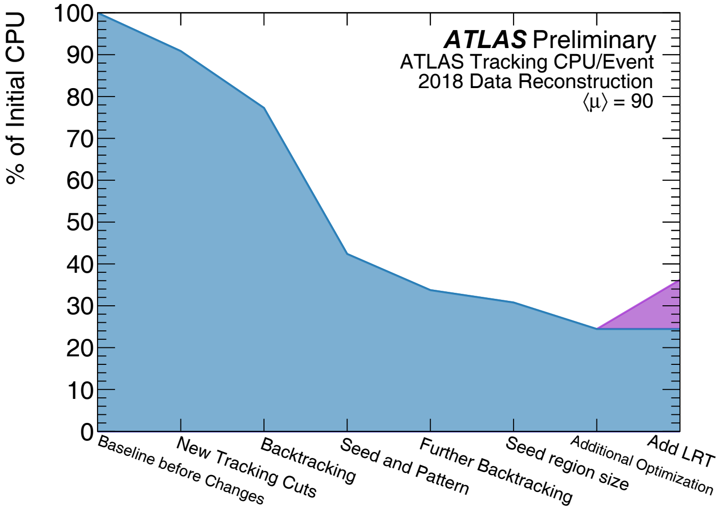


Figure 1. Incremental decrease of the CPU time taken to reconstruct a set of $\langle\mu\rangle = 90$ events when adding improvements to the Run 3 track reconstruction. The blue shaded area indicates the time, relative to the initial Run 3 software implementation, taken for the tracking passes that were also run during Run 2. The purple area indicates the time added by the additional LRT pass. Taken from reference [6].

4 Benchmarking Methodology

The impact of the improvements discussed in the previous sections on the software and reconstruction performance is evaluated using recorded collision data and Monte-Carlo (MC) simulated $t\bar{t}$ samples. For various pile-up values, the raw data of sets of 300 consecutive collision events are reconstructed, and processing time taken for the reconstruction as well as the size of the output written to disk is recorded. The samples are taken from a single LHC run (fill number 6291) recorded towards the end of the 2017 data-taking campaign and covering a range of pile-up values between $\langle\mu\rangle = 15.5$ and 60. This ensures consistent data-taking conditions across all pile-up values. All the events taken from this run fall under the so-called good-run list (GRL), meaning that they satisfy all data quality requirements to be considered part of the ATLAS physics dataset. To extend the study towards even larger values of pile-up of up to $\langle\mu\rangle = 90$, an LHC run recorded in late 2018 (fill 7358) as part of a machine-development campaign is used in addition. Unlike the 2017 data, this run is not considered part of the ATLAS physics dataset due to the nonstandard data-taking conditions. Since the ID was fully operational, it is however possible to use the respective events to obtain an estimate of the scaling behaviour of track reconstruction performance under extreme pile-up conditions.

All benchmarks described in this work were run as the only active user on a dedicated machine equipped with two Intel(R) Xeon(R) E5-2630 v3 8-core, 2.4 GHz processors and 128 GB of RAM, running the CERN CENTOS 7 operating system. CPU scaling was set to performance mode and hyper-threading disabled. A HS06 score of 278 is reported [17] for this processor without hyperthreading. The machine was kept at a stable 50% in capacity by running an appropriate number of reconstruction tasks simultaneously. To exclude the impact of multi-threading from the comparison all tests were run in single-thread mode.

The binaries tested in this work are identical to those most commonly used for the experiment's regular data reconstruction. As a result, different versions of the software differ not only in terms of their own programming, but also in terms of the method of compilation as well as external libraries and the compilation thereof. All versions of the software were compiled with the default compilation settings as they were defined in the ATLAS software project at the time of their release. Since the optimisation flags are partially set according to compiler presets, the exact details may differ between compiler versions. In all cases, however, the code is compiled for a generic x86-64 architecture, implying support for vector instruction set extensions up to SSE2. The binaries produced are therefore unable to exploit more modern architectural features like AVX. The binaries for the Run 2 reconstruction were compiled using version 6.2.0 of the GNU Compiler Collection, whereas the Run 3 binaries utilise version 8.3.0 of *gcc*.

5 Performance Results

The comparison of the time taken for track reconstruction in the software release used during Run 2 data taking versus the new release prepared for Run 3 is shown in Figure 2 as a function of $\langle\mu\rangle$. The green curve shows the timing requirement for track reconstruction using the Run 2 release, the purple indicates the timing for the Run 3 release, while the blue shows the impact of including the LRT secondary pass in the default reconstruction chain. In both cases the performance for the new release is more than a factor of 2 faster. Near linear scaling of the CPU consumption with $\langle\mu\rangle$ is now observed compared to the behaviour seen for Run 2.

A breakdown of the speed-up seen for the individual major parts of track reconstruction, as described in Section 2, is shown in Figure 3. For Run 2 the track-finding step (violet) was by far the largest CPU consumer for track and total ATLAS reconstruction and scaled non-linearly with $\langle\mu\rangle$. This behaviour has been rectified and the timing of the pattern recognition has been reduced up to a factor of 4. Nearly all of the major consumers see a reduction of around a factor of 1.5 to 2.0.

A comparison between the Run 2 and Run 3 track reconstruction for absolute values of time required per event is illustrated in Figure 4. The secondary pass added to Run 3 reconstruction is indicated in the white area superimposed on top for the Run 3 part of the figure. The violet area shows the dramatic reduction of absolute time required for the track finding part, with respect to Run 2. The $\langle\mu\rangle$ dependency observed for Run 2 is closer to linear after the optimisations, showing that the Run 3 tracking software is well prepared for high $\langle\mu\rangle$ data-taking. Similarly, Figure 5 shows the fraction of total ID reconstruction taken by components for Run 2 and Run 3 reconstruction.

Track reconstruction accounted for around 64% of the total ATLAS event reconstruction CPU time in Run 2. This fraction has been reduced to 40% for Run 3 at $\langle\mu\rangle = 50$. The breakdown of the total time between the different domains of reconstruction is shown in Figure 6.

Storage capacity is also a limited commodity and heavily challenged by the vast amounts of collision data to be recorded. With the Run 3 track reconstruction improvements, fake track reconstruction rates have been drastically reduced, and the average quality of the tracks

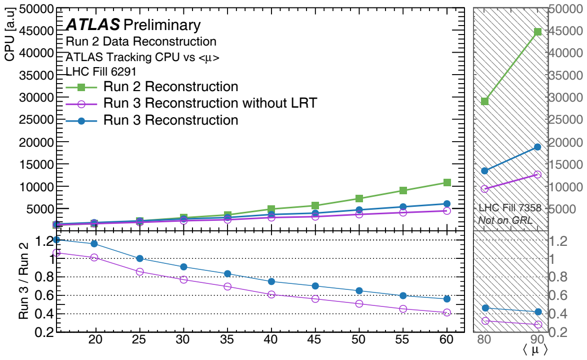


Figure 2. Processing time taken per event versus average pileup to reconstruct the same data events, comparing the Run 2 (green) and Run 3 (purple) reconstruction software. The Run 3 numbers are also presented including the impact of the additional LRT tracking pass (blue). The bottom panel depicts the time taken as a fraction of the Run 2 result. The shaded area indicates data events taken from a 2018 machine development run not passing the full ATLAS data quality requirements. Taken from reference [6].

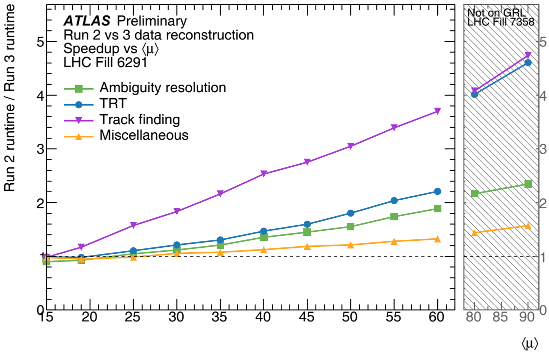
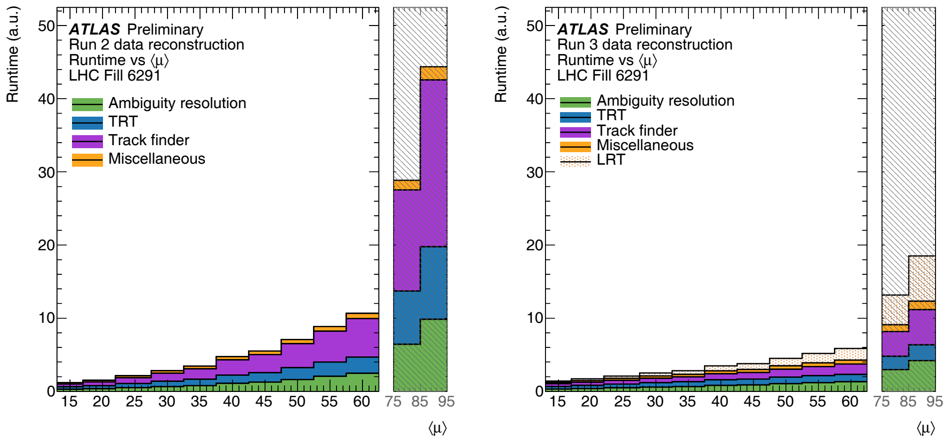


Figure 3. Breakdown of the speed improvement in the Run 3 software compared to the Run 2 iteration for key components of the track reconstruction as a function of $\langle\mu\rangle$. The shaded area indicates data events taken from a 2018 machine development run not passing the full ATLAS data quality requirements. Taken from reference [6].

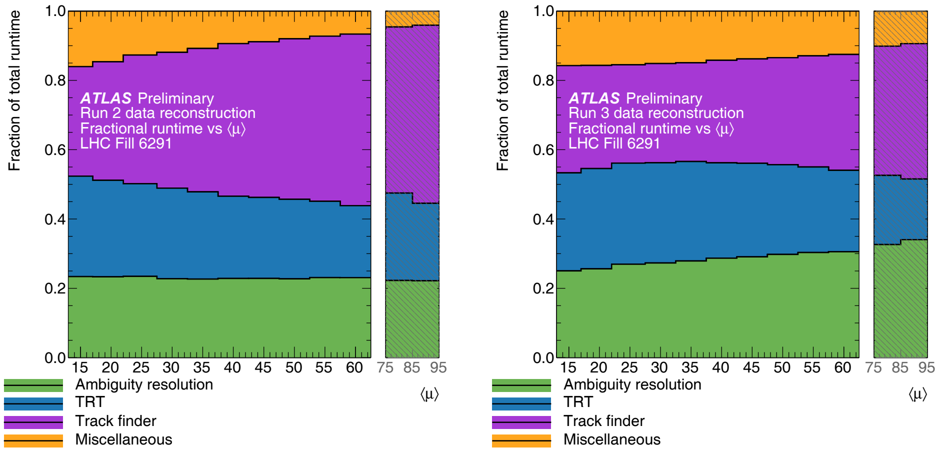
has increased. This surmounts to a large reduction in the overall number of output tracks written to disk, reducing the needs for storage space. Even after including the additional tracks from the LRT a reduction of up to 50% is achieved at the highest pile-up values. The output size for tracks in kilobytes per event is shown for the standard ATLAS event data format in Figure 7, and illustrates up to a $20-50\%$ reduction in the required disk space. Additionally, the scaling with pile-up has been significantly reduced in the Run 3 release, leading to larger improvements at higher values of $\langle\mu\rangle$.



(a)

(b)

Figure 4. Breakdown of CPU consumer for track and vertex reconstruction comparing Run 2 (a) and Run 3 (b) configurations versus $\langle \mu \rangle$, shown in absolute units. Taken from reference [6].

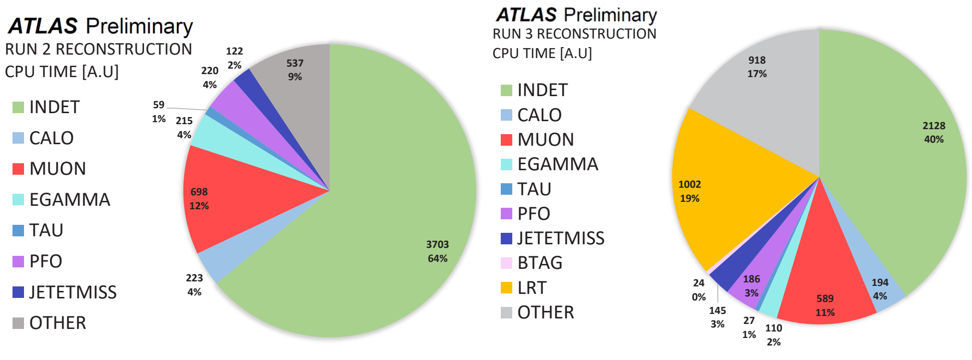


(a)

(b)

Figure 5. Breakdown of CPU consumer for track and vertex reconstruction comparing Run 2 (a) and Run 3 (b) configurations versus $\langle \mu \rangle$, shown as the fraction of the total runtime for the individual components. Taken from reference [6].

Finally, the software improvements have not negatively impacted the track reconstruction physics performance compared to Run 2. The tracking efficiency, defined as the fraction of charged particles originating from the primary *p*-*p* interaction that were successfully reconstructed, is shown for the Run 2 and Run 3 reconstruction in Figure 8(a) as a function for the truth particle transverse momentum. The efficiency loss is smaller than 4% at low p_T and smaller than 1% at larger transverse momenta. The slight reduction compared to Run 2 is a



(a) (b)

Figure 6. Fraction of the total CPU requirement for full ATLAS reconstruction split by domain for Run 2 (a) and Run 3 (b) for one data run at $\langle\mu\rangle = 50$. The slices are defined as follows: INDET: Inner Detector track and vertex reconstruction. CALO: Preprocessing and clustering of cells in the Tile and LAr calorimeter. MUON: Muon spectrometer track reconstruction and ID combined muons. EGAMMA: Dedicated electron track reconstruction, γ-conversion secondary vertex finding, electron- and γ-object reconstruction. TAU: Tau reconstruction. PFO: Charged and neutral particle flow jet reconstruction. JETETMISS: Initial jet and missing E_T reconstruction. BTAG: Low level flavour tagging reconstruction for monitoring. LRT: Summed contribution for all domains for producing displaced objects. OTHER: Data-writing, and isolation building. Taken from reference [6].

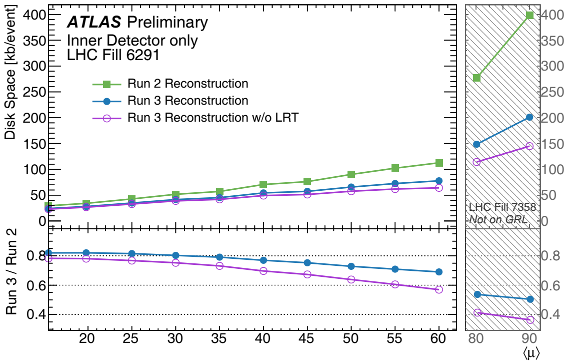
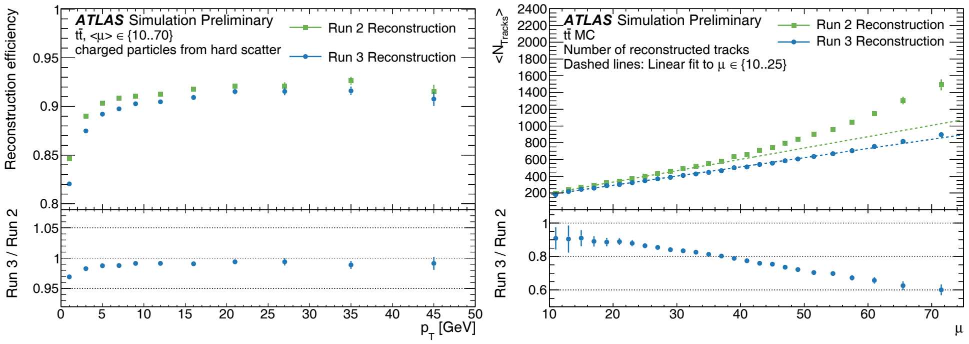


Figure 7. Event size of the Inner Detector reconstruction output in the ATLAS Event data format for the same set of reconstructed data events as a function of average pile-up, comparing the Run 2 and Run 3 releases. The shaded area indicates data events taken from a 2018 machine development run not passing the full ATLAS data quality requirements. Taken from reference [6].

result of the stricter requirements on the number of silicon clusters and impact parameter for a track to be retained, and is not expected to affect the reconstruction of muons as minimum ionising particles.

A measure of the rate of incorrect combinations of clusters reconstructed as tracks is the number of tracks per event as a function of $\langle\mu\rangle$. The number of real tracks is expected to scale linearly with this quantity, since it is related to the number of charged particles produced in the collisions. The number of random combinations is expected to scale with a higher power, since the enhanced combinatorics allow for more such candidates to be formed. The mean number of reconstructed tracks as a function of $\langle\mu\rangle$ is shown in Figure 8(b). To give an impression of the linearity, a linear fit to the $5 < \langle\mu\rangle < 20$ is superimposed as a dashed line. While a clear non-linear component amounting to up to 30% of the total number of tracks is visible for the Run 2 reconstruction, the Run 3 reconstruction is observed to show a close to ideally linear behaviour, demonstrating the dramatic improvement in purity of the reconstructed tracks.



(a) (b)

Figure 8. Tracking efficiency as a function of p_T (a) and mean number of reconstructed tracks per event as a function of $< \mu >$ (b) in simulated $t\bar{t}$ events with a $\langle\mu\rangle$ distribution matching the conditions of LHC Run 2, comparing Run 2 and Run 3. Taken from reference [6].

6 Conclusions

The upcoming LHC Run 3 represents a major challenge to the experiments' event reconstruction. In ATLAS, a major improvement of the inner detector track reconstruction software has been performed in order to ensure that reconstruction remains feasible within the available computing resources. An execution speed improvement between a factor of 2 and 4, depending on the pile-up, is achieved. These improvements allow the execution of an additional tracking pass benefiting long-lived particle searches while still retaining a significant overall performance improvement compared to the past software iteration. This ensures that ATLAS will be able to efficiently reconstruct collision data in LHC Run 3.

References

[1] ATLAS Collaboration, JINST **3**, S08003 (2008)
[2] ATLAS Collaboration, ATLAS-TDR-19 (2010)
[3] Abbott, B. and others, JINST **13**, T05008 (2018)
[4] ATLAS Collaboration, ATLAS-CONF-2019-021 (2019)
[5] ATLAS Collaboration, ATL-PHYS-PUB-2021-012 (2021)
[6] ATLAS Collaboration, ATL-PHYS-PUB-2021-172 (2021)
[7] ATLAS Collaboration, ATL-PHYS-PUB-2015-051 (2015)

[8] ATLAS Collaboration, ATL-PHYS-PUB-2018-002 (2018)

[9] ATLAS Collaboration, ATL-PHYS-PUB-2015-031 (2015)

[10] ATLAS Collaboration, Eur. Phys. J. C **77**, 673 (2017)

[11] ATLAS Collaboration, Eur. Phys. J. C **80**, 1194 (2020)

[12] R. Frühwirth, Nucl. Instrum. Methods Phys. Res. A **262**, 0168-9002 (1987)

[13] ATLAS Collaboration, Eur. Phys. J. C **77**, 332 (2017)

[14] ATLAS Collaboration, ATL-PHYS-PUB-2017-014 (2017)

[15] ATLAS Collaboration, ATL-PHYS-PUB-2019-015 (2019)

[16] A. Salzburger et al., Acts project (2020)

[17] HEPiX Benchmarking Working Group, HEP-SPEC06 Results for SL6 x86_64 (gcc 4.4) Benchmark Environment