PROCEEDINGS OF SCIENCE

Tracking, Vertexing and data handling strategy for the LHCb upgrade

Paul Seyfert[*][†]
CERN
E-mail: Paul.Seyfert@cern.ch

For Run III (2021 onwards) of the LHC, LHCb will take data at an instantaneous luminosity of $2 \times 10^{33}\mathrm{cm}^{-2}\mathrm{s}^{-1}$, five times higher than in Run II (2015-2018). To cope with the harsher data taking conditions, the LHCb collaboration will upgrade the DAQ system and install a purely software based trigger, in addition to various detector upgrades. The high readout rate contributes to the challenge of reconstructing and selecting events in real time.

Special emphasis in this contribution will be put on the need for fast track reconstruction in the software trigger. The modified detector infrastructure will be able to face this challenge and the necessary changes to the reconstruction sequence are discussed. A novel strategy is presented which distributes and maximises the bandwidth among the different physics channels using a genetic algorithm.

The data processing chain includes a redesign of the event scheduling, introduction of concurrent processing, optimisations in processor cache accesses and code vectorisation. Furthermore changes in the areas of event model, conditions data and detector description are foreseen.

The 26th International Workshop on Vertex Detectors
10-15 September, 2017
Las Caldas, Asturias, Spain

[*] Speaker.

[†] On behalf of the LHCb collaboration.

1. Motivation for an LHCb upgrade

While originally mainly motivated for a small number of b hadron decay analyses [1], the LHCb experiment has proven its capabilities as a multi-purpose detector in roughly 400 physics analysis papers. The success is to significant part thanks to the ability to operate beyond the design pile-up [2], the deployment of real-time calibration and alignment [3], and the application of real-time analysis methods [4].

Despite having collected more data than anticipated, the achieved experimental precision on many benchmark modes does not reach that of theoretical predictions, and many experimental results will remain statistically limited by the end of the LHC's second run in 2018. This means that larger data samples will not only improve the experimental sensitivity, but also allow for theoretical interpretation [5, 6].

A significant increase of the data sample can be obtained with an increase of the instantaneous luminosity to $\mathscr{L} \approx 2 \times 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. This is within the possibilities of the current LHC without the upgrades to the accelerator scheduled for the operations of the HL-LHC in Run IV. The detector will receive major upgrades and replacements including a new vertex pixel detector [7], a new silicon strip detector (UT) [8], and a scintillating fibre tracker [9]. While reliably operating at a higher track density, these upgrades must maintain the current detector's resolution – otherwise the gain in statistical sensitivity through luminosity increase would be lost in statistical dilution from the detector resolution effects.

2. Design requirements and implications for data taking

The goals set for the physics reach of the upgraded LHCb experiment impose strong requirements on the data taking strategy. A core aspect is the so-called trigger-less readout, where the entire event selection happens in software, without hardware trigger preselection, and without tight latency requirements on single events' processing times.

2.1 Full software trigger

At the current LHCb detector with a fixed detector readout rate, an increase in instantaneous luminosity must be accounted for in the trigger. For signatures that are triggered with the calorimeters, it is expected that in Run III backgrounds will become indistinguishable after a certain E_T threshold, resulting in a random rejection of events which cancels the increased production rate of signal signatures, as illustrated in Fig. 1. It is therefore not possible to achieved the goals of the LHCb upgrade with a hardware trigger and a detector readout rate lower than $\mathscr{O}(30\,\mathrm{MHz})$. Another reason LHCb wants to perform all event selections in software is that some relevant signatures can only be identified by means of the RICH particle identification, e.g. charm decays are expected to occur at MHz rates [10] and "interesting" charm decays are typically suppressed with respect to Cabibbo favoured modes, and can only be distinguished by means of K/π separation of the decay products.

2.2 Real-time analysis

The analysis of high rate signals is already challenging for the budget of the LHCb offline

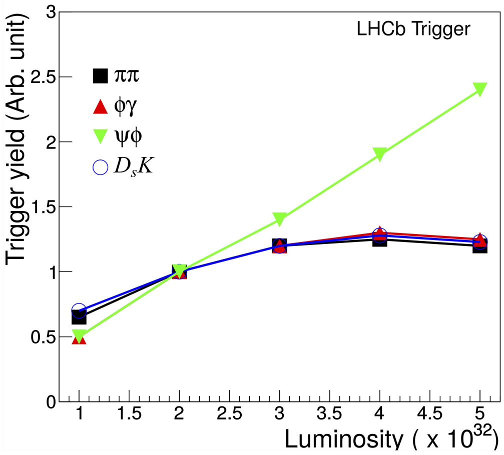


Figure 1: Increase in signal yield after trigger as function of the instantaneous luminosity for a fixed detector readout bandwidth, controlled through hardware trigger thresholds of the existing LHCb detector. Yields for channels with muonic triggers increase proportionally to the luminosity. The increased thresholds for calorimeter triggers compensates the increased production rate and leads to a saturation of signal yields.

computing resources. In Run III, events must be analysed in real time[1]: once an event is completely reconstructed in the software trigger for the event selection, only high level quantities (e.g. decay kinematics) and no detector raw data are stored and a second reconstruction offline is not possible. This “Turbo” approach [4], already deployed for some selections in Run II, will become the default in the LHCb upgrade.

The readout scheme is not supposed to suit a “one size fits all” paradigm. It becomes the analysts’ duty to define, before data taking, which event data is needed for the analysis. Throughout Run II, the selective storage has been deployed in stages. The extremes are to only store the relevant reconstructed decay and nothing else [11], or storing all reconstructed objects in an event [12]. Since 2017, a fine granulated selection of objects to store has been deployed where analysts can define if they need e.g. all charged tracks from a single primary vertex or within a cone around their signal, which will enable offline flavour tagging.

2.3 Real-time calibration

To profit from optimal reconstruction quality in real-time analyses, LHCb must continue its real-time alignment calibration as deployed in Run II [3]. The readout scheme envisaged for Run III is outlined in Fig. 2: the detector readout happens at $\mathscr{O}(30\,\text{MHz})$, the rate of inelastic events, and event building is done for all events, resulting in a data rate of $\mathscr{O}(40\,\text{Tbit/s})$. Assuming the same overall high level trigger (HLT) layout as in the current LHCb setup, these events will be processed immediately by a first stage (HLT1), and can then be buffered. That buffer can – depending on the retention in HLT1 – hold events for days and thereby lifts latency requirements for all other steps in the processing chain.

[1] Real time is not a latency requirement here, as discussed in Sect. 2.3.

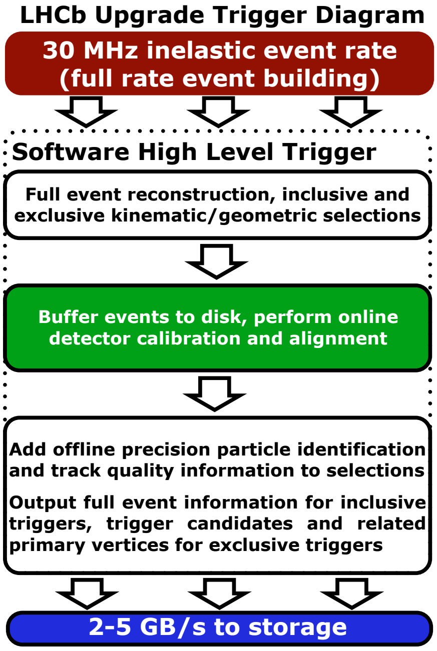

LHCb Upgrade Trigger Diagram
30 MHz inelastic event rate (full rate event building)
Software High Level Trigger
Full event reconstruction, inclusive and exclusive kinematic/geometric selections
Buffer events to disk, perform online detector calibration and alignment
Add offline precision particle identification and track quality information to selections
Output full event information for inclusive triggers, trigger candidates and related primary vertices for exclusive triggers
2-5 GB/s to storage

Figure 2: Readout scheme of the LHCb upgrade trigger.

Calibration and alignment tasks are expected to run at the beginning of an LHC fill within the current time budget of a few minutes. From that point onwards, all events entering HLT1 are processed with the latest and best calibration and alignment constants. Events which have been buffered until that point could also be processed with updated constants in the second HLT stage (HLT2), however for consistency of the two reconstruction stages, it has been chosen to use the same constants for any given event in both HLT stages.

2.4 Bandwidth division

Even if the demand for offline storage is reduced by the real-time analysis strategy (with respect to storing all raw data), further reduction of the trigger output bandwidth is needed. In addition to interesting signal events, every event selection will select backgrounds and every analyst should be able to make trade-offs between signal efficiency and signal purity (and thus data retention). These trade-offs can be assumed to be made as optimal as achievable if every event selection is done with a machine learning algorithm, providing a single numeric probability that an event is a signal event, which can then be used for the trigger decision.

The trigger for all research areas of LHCb is then set up with a genetic algorithm [13]. It should establish a fair share between all selections. This global optimisation is allowed to vary the selection requirement on the machine learning classifier response of every event selection. The optimisation minimises an overall χ^2 defined as

$$\chi^2 := \sum_i w_i \left(1 - \frac{\varepsilon_i}{\varepsilon_i^{\max}}\right)^2 ,$$

where the sum ranges over all selections, ε_i is the selection efficiency for selection i, which gets normalised to $\varepsilon_i^{\max}$, the maximal possible signal efficiency for selection i if this selection would be allowed to use the entire LHCb trigger bandwidth for its own. The weight factor w_i allows to account for human judgement of how important a selection is. For example a lower relevance can be assigned to high rate signals which are used for normalisation or calibration purposes (if reduced statistics would not lead to a reduced sensitivity).

If during the optimisation procedure, a trigger configuration exceeds the total available bandwidth, all efficiencies are reduced by the factor by which the bandwidth is exceeded; this emulates what random dropping of events in the data acquisition would result in, which would be the last resort during data taking. Such random dropping should always perform worse in the figure of merit than a more restrictive selection requirement with the machine learning classifier, and thus it is expected that this does not occur in the final configuration.

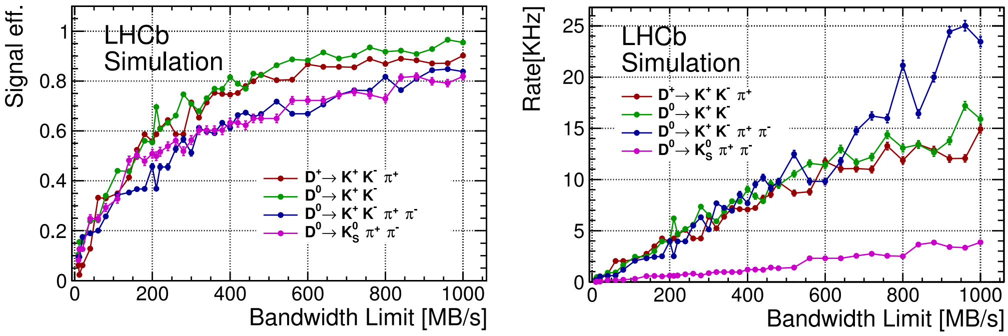


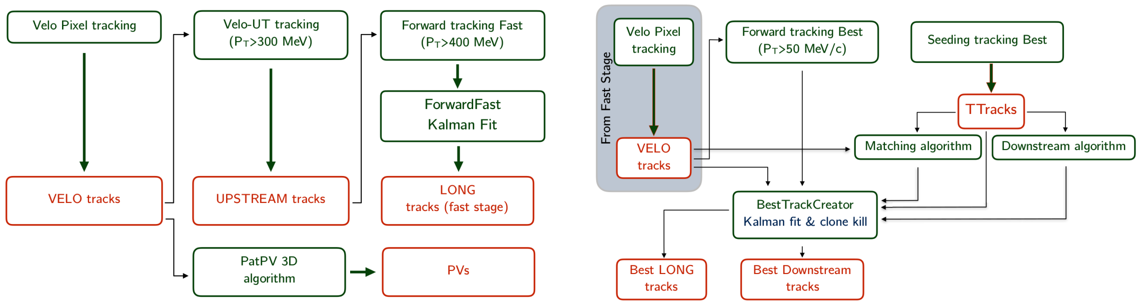
Figure 3: Example for a bandwidth division between four charm meson decay selections from Ref. [13]. For bandwidth limits between 400 MB/s and 1000 MB/s the efficiency is rather flat – rate reduction is achieved through background rejection. For the $D^0 \to K_S^0\pi^+\pi^-$ selection (magenta), the candidate and all other reconstructed objects are stored, for all other selections only the decay candidate is stored. This leads to the difference in rate between the $D^0 \to K_S^0\pi^+\pi^-$ and the other selections.

Initial studies on a small set of example selections, shown in Fig. 3, shows that a reduction of trigger output bandwidth by a factor of two, leads to a efficiency reduction of only 10%. As expected from a tuneable selection, most of the bandwidth is saved by rejecting background events.

This bandwidth division is following LHCb's current strategy of adjusting the hardware trigger settings. One of the aims is to have stable settings throughout data taking to provide homogeneous thresholds. Similarly, the trigger configuration shouldn't be changed on a fill-to-fill basis, but adjustments, e.g. to account for unexpectedly good accelerator performance, are anticipated and welcome.

3. Tracking

One of the most important parts of the software trigger at LHCb is and will be the track reconstruction. Similar to the current LHCb software trigger, a division into two stages is planned [14]. The tracking sequences for both stages are outlined in Fig. 4. The first stage should reduce the rate by a factor ~ 30 and apply single track and two track selections for displaced signatures. Such selections with small numbers of tracks scale well at high detector occupancies and – since not all decay products of an interesting decay are needed – reach high efficiencies even at elevated momentum thresholds. The remaining tracks are then reconstructed in the second trigger stage with lower momentum thresholds and additional algorithms to reconstruct decays of strange and long-lived particles happening outside the vertex locator.



(a) Fast tracking sequence for the first stage of the software trigger.

(b) Full tracking sequence for the second stage of the software trigger.

Figure 4: Schematic tracking sequences for the software trigger in the LHCb upgrade. [14]

3.1 Computational challenge

For the design of the upgraded trigger CPU farm, Moore's law, the increase of processing power of CPUs per cost unit, must be taken into account. Yet, comparing CPUs from different years of acquisition in the LHCb trigger farm, the growth in raw computational power (in GFLOPS) is a factor 10 larger than the growth in trigger throughput (in trigger decisions per second).

This shortcoming of newer and better CPUs, to compute trigger decisions faster, is largely due to the lack of parallelism in the current LHCb software. The growth of processing power in modern CPUs is partially due to the proliferation of SIMD[2] units, that can perform the same operation simultaneously on multiple inputs. Software not designed for that leaves potential processing power in these units unused. The availability of multiple cores on a computer, in addition to hyperthreads, is currently used in the LHCb trigger operations by launching multiple processes instead of parallelising the software. These processes then compete for the same CPU caches and suffer from the fact that growth in memory bandwidth does not match that of CPU performances. This is especially unfortunate since a large part of the memory usage is not event data, but data that is identical for all processes, such as the detector geometry.

3.2 Track fit improvements

An important part of the speedup of the reconstruction is a reimplementation of the Kalman filter track fit [16] – one of the largest CPU time consumers. The Kalman filter itself is a sequential algorithm, but fits of different tracks are independent of each other and it is therefore parallelisable on the track level. Furthermore, the same matrix operations are performed for all tracks and the parallelisation can therefore even be performed on an SIMD unit. The matrix algebra steps of the fit have been sped up by a factor 2 [17, 18] by using code vectorisation. This speedup is reduced

[2] single instruction multiple data, according to Flynn's taxonomy [15]

by other parts of the track fit that remain unvectorised such as the material lookup. Furthermore, vectorisation requires a modified memory layout of the event data. Traditionally, object oriented programming leads to data layout as arrays of structures, which are widely used in the LHCb software. Vectorisation however requires data layout as structure of arrays (SOA). The back-and-forth conversion costs additional time. For the future, it is explored whether changing the common data layout into SOA structures, and thereby avoiding conversions, is feasible to deploy for LHCb's entire software stack.

The track fit relies on extrapolating charged particles' trajectories through the magnetic field. These are commonly done from first principles of electrodynamics and Runge-Kutta methods, although effective models can reach the same accuracy,[3] as illustrated in Fig. 5. The lookup of material densities is reduced to $\mathcal{O}(20)$ parameters and yields the same momentum resolution as the detailed detector description for particle momenta above $\sim 7\,\mathrm{GeV}/c$. It is still under investigation if alternative parametrisations can reach the nominal fit resolution near the lower momentum threshold of the reconstruction, or if separate extrapolation models for fits at very low momenta will be deployed in Run III – Ultimately the goal is to maintain the best possible resolution at a small computational foot print.

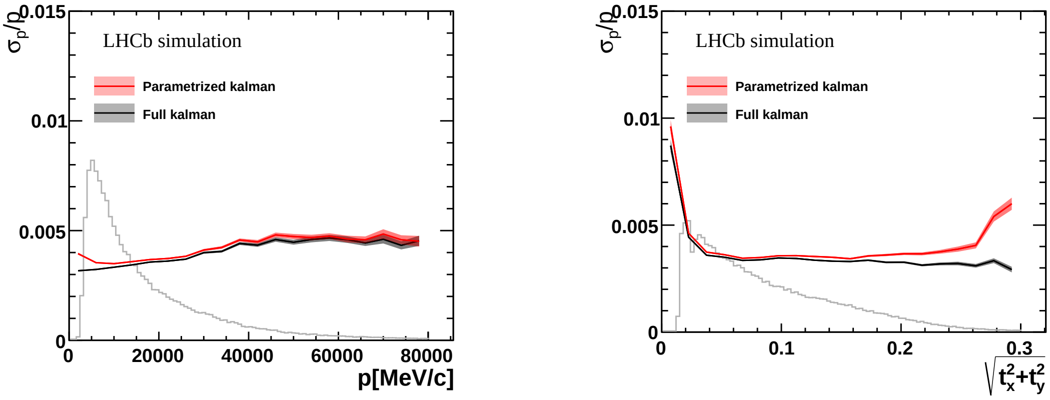


Figure 5: Momentum resolution of a parametrised Kalman filter fit to the nominal (full) Kalman filter fit. Above $p \sim 7\,\mathrm{GeV}/c$ and except for small pseudo rapidity (above azimuthal angles of $\sqrt{t_x^2+t_y^2} := \tan\theta \sim 0.2$) both models have the same resolution. The grey histogram shows the momentum and azimuthal distribution of tracks in the LHCb acceptance.

3.3 Fake track identification

Fake tracks can largely contribute to the consumption of computing resources, since the reconstruction of a decay is a combinatorial task of reconstructed tracks. Following the Run II strategy, a neural network is used to distinguish real particles' tracks from reconstruction artefacts after the track fit [19]. The neural network performs better than the track fit χ^2. In Run II, this network reduces the CPU consumption for decay combinatorics by $\mathcal{O}(60\,\%)$ and reduces the trigger output rate by $\mathcal{O}(30\,\%)$. The network will receive further updates and retrainings in the future but its

[3] These results have been updated with respect to what was shown at the conference, significantly improving the resolution of the parametrised fit.

relevance is likely to decrease becasue more machine learning enters the pattern recognition before the track fit, which reduces the prevalence of fake tracks already before the track fit [20].

4. Functional processing framework

A massive reduction in memory consumption is reached by changing to a functional processing framework which facilitates parallel processing. In current benchmarks, the speedup from single threaded applications to multithreaded applications is very promising. On a 64 core benchmark machine with 64 application threads a 56-fold speedup is observed (12 % missing with respect to perfect speedup). At the same time, the memory consumption only increases 2.5-fold.

Converting the software stack to the new functional processing framework was not implemented by the core software developers alone. An important contribution were hackathons. At four to five of these per year, subsystem experts were trained in the usage of the framework and assisted in converting existing algorithms and establishing thread safety.

5. Conclusion

The physics objectives of the LHCb upgrade require reading out the detector at the full rate of inelastic collisions of 30 MHz. The event selection will be done entirely in software where the resolution of the reconstruction must not be deteriorated to improve the processing speed.

The limitations of computing resources puts tight constraints on the reconstruction and storage resources, this makes it necessary to optimise the use of CPU and storage. To this goal, the reconstruction outcome of the trigger will be stored and analysed directly rather than relying on a subsequent offline reconstruction. Without offline reconstruction, it is not necessary to store raw event data anymore, and every selection will need to specify which data will be needed for offline analysis. Optimal usage of modern computer architectures requires to tailor the reconstruction software to thread safe multi threaded processing and to usage of SIMD units.

References

[1] LHCb collaboration, B. Adeva et al. “Roadmap for selected key measurements of LHCb”. LHCb-PUB-2009-029 (2009). arXiv: 0912.4179.

[2] LHCb collaboration, R. Aaij et al. “LHCb detector performance”. *Int. J. Mod. Phys.* A30 (2015), p. 1530022. DOI: 10.1142/S0217751X15300227. arXiv: 1412.6352.

[3] LHCb collaboration, S. Borghi. “Novel real-time alignment and calibration of the LHCb detector and its performance”. *14th Vienna Conference on Instrumentation (VCI 2016). Nucl. Instrum. Meth.* Vol. A845 (2017), pp. 560–564. DOI: 10.1016/j.nima.2016.06.050.

[4] S. Benson et al. “The LHCb Turbo Stream”. *21st International Conference on Computing in High Energy and Nuclear Physics (CHEP 2015). J. Phys. Conf. Ser.* Vol. 664.8 (2015), p. 082004. DOI: 10.1088/1742-6596/664/8/082004.

[5] LHCb collaboration, R. Aaij et al. “Letter of Intent for the LHCb Upgrade”. LHCC-I-018 (2011). URL: https://cds.cern.ch/record/1333091.

[6] LHCb collaboration, R. Aaij et al., and A. Bharucha et al. “Implications of LHCb measurements and future prospects”. *Eur. Phys. J.* C73 (2013), p. 2373. DOI: 10.1140/epjc/s10052-013-2373-2. arXiv: 1208.3355.

[7] E. L. Cid. “The LHCb Vertex Locator Upgrade”. *Proceedings of the 26th International Workshop on Vertex Detectors (VERTEX)*. Ed. by I. Vila et al. 2017.

[8] M. Petruzzo. “Design and construction of the LHCb Upstream Tracker”. *Proceedings of the 26th International Workshop on Vertex Detectors (VERTEX)*. Ed. by I. Vila et al. 2017.

[9] LHCb collaboration, A. A. Alves Jr. et al. “LHCb Tracker Upgrade Technical Design Report”. LHCB-TDR-015 (2014). URL: http://cds.cern.ch/record/1647400.

[10] C. Fitzpatrick and V. V. Gligorov. “Anatomy of an upgrade event in the upgrade era, and implications for the LHCb trigger”. LHCb-PUB-2014-027 (2014). URL: https://cds.cern.ch/record/1670985.

[11] LHCb collaboration, R. Aaij et al. “Measurements of prompt charm production cross-sections in pp collisions at $\sqrt{s} = 13$ TeV”. *JHEP* 03 (2016). [Erratum: JHEP05,074(2017)], p. 159. DOI: 10.1007/JHEP03(2016)159. arXiv: 1510.01707.

[12] LHCb collaboration, R. Aaij et al. “Study of J/ψ Production in Jets”. *Phys. Rev. Lett.* 118.19 (2017), p. 192001. DOI: 10.1103/PhysRevLett.118.192001. arXiv: 1701.05116.

[13] C. Fitzpatrick et al. “Upgrade trigger: Bandwidth strategy proposal”. LHCb-PUB-2017-006 (2017). URL: http://cds.cern.ch/record/2244313.

[14] R. Aaij et al. “Upgrade trigger: Biannual performance update”. LHCb-PUB-2017-005 (2017). URL: https://cds.cern.ch/record/2244312.

[15] M. J. Flynn. “Some Computer Organizations and Their Effectiveness”. *IEEE Transactions on Computers* C-21.9 (1972), pp. 948–960. ISSN: 0018-9340. DOI: 10.1109/TC.1972.5009071.

[16] R. Frühwirth. “Application of Kalman filtering to track and vertex fitting”. *Nucl. Instrum. Meth.* A262 (1987), pp. 444–450. DOI: 10.1016/0168-9002(87)90887-4.

[17] D. H. Cámpora Pérez. “LHCb Kalman Filter cross architecture studies”. *22nd International Conference on Computing in High Energy and Nuclear Physics (CHEP 2016). J. Phys. Conf. Ser.* Vol. 898.3 (2017), p. 032052. DOI: 10.1088/1742-6596/898/3/032052.

[18] D. H. Cámpora Pérez, O. Awile, and C. Potterat. “A High-Throughput Kalman Filter for Modern SIMD Architectures”. *Euro-Par 2017: Parallel Processing Workshops*. Springer International Publishing, pp. 378–389. DOI: 10.1007/978-3-319-75178-8_31.

[19] M. De Cian et al. “Fast neural-net based fake track rejection”. LHCb-PUB-2017-011 (2017). URL: http://cds.cern.ch/record/2255039.

[20] LHCb collaboration, M. Stahl. “Machine learning and parallelism in the reconstruction of LHCb and its upgrade”. *22nd International Conference on Computing in High Energy and Nuclear Physics (CHEP 2016). J. Phys. Conf. Ser.* Vol. 898.4 (2017), p. 042042. DOI: 10.1088/1742-6596/898/4/042042. arXiv: 1710.08947.