

Tesla: An application for real-time data analysis in High Energy Physics[☆]



R. Aaij^a, S. Amato^b, L. Anderlini^c, S. Benson^{a,*}, M. Cattaneo^a, M. Clemencic^a,
B. Couturier^a, M. Frank^a, V.V. Gligorov^d, T. Head^e, C. Jones^f, I. Komarov^e, O. Lupton^g,
R. Matev^a, G. Raven^h, B. Sciasciaⁱ, T. Skwarnicki^j, P. Spradlin^k, S. Stahl^a, B. Storaci^l,
M. Vesterinen^m

^a European Organization for Nuclear Research (CERN), Geneva, Switzerland

^b Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil

^c Sezione INFN di Firenze, Firenze, Italy

^d LPNHE, Université Pierre et Marie Curie, Université Paris Diderot, CNRS/IN2P3, Paris, France

^e Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

^f Cavendish Laboratory, University of Cambridge, Cambridge, United Kingdom

^g Department of Physics, University of Oxford, Oxford, United Kingdom

^h Nikhef National Institute for Subatomic Physics and VU University Amsterdam, Amsterdam, The Netherlands

ⁱ Laboratori Nazionali dell'INFN di Frascati, Frascati, Italy

^j Syracuse University, Syracuse, NY, United States

^k School of Physics and Astronomy, University of Glasgow, Glasgow, United Kingdom

^l Physik-Institut, Universität Zürich, Zürich, Switzerland

^m Physikalisches Institut, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany

ARTICLE INFO

Article history:

Received 3 May 2016

Received in revised form

4 July 2016

Accepted 11 July 2016

Available online 17 July 2016

Keywords:

Particle physics

Data analysis

Data acquisition

ABSTRACT

Upgrades to the LHCb computing infrastructure in the first long shutdown of the LHC have allowed for high quality decay information to be calculated by the software trigger making a separate offline event reconstruction unnecessary. Furthermore, the storage space of the triggered candidate is an order of magnitude smaller than the entire raw event that would otherwise need to be persisted. Tesla is an application designed to process the information calculated by the trigger, with the resulting output used to directly perform physics measurements.

© 2016 Published by Elsevier B.V.

1. Introduction

The LHCb experiment, one of the four main detectors situated on the Large Hadron Collider in CERN, Geneva, specialises in precision measurements of beauty and charm hadrons decays. At the nominal LHCb luminosity of $4 \times 10^{32} \text{ cm}^{-2} \text{ s}^{-1}$ during 2012 data taking at 8 TeV, around 30 k beauty (b) and 600 k charm (c) hadron pairs pass through the LHCb detector each second. Each recorded collision is defined as an event that can possibly contain a decay of

interest. The efficient selection of beauty and charm decays from the ~ 30 M proton–proton collisions per second is a significant Big Data challenge.

An innovative feature of the LHCb experiment is its approach to Big Data in the form of the High Level Trigger (HLT) that is split into two components with a buffer stage between [1]. The HLT is a software application designed to reduce the event rate from 1 M to ~ 10 k events per second and is executed on an Event Filter Farm (EFF). The EFF is a computing cluster consisting of 1800 server nodes, with a combined storage space of 5.2 PB, which can accommodate up to two weeks of LHCb data taking [2] in nominal conditions. The HLT application reconstructs the particle trajectories of the event in real time, where real time is defined as the interval between the collision in the detector and the moment

[☆] © CERN on behalf of the LHCb collaboration, licence CC-BY-4.0.

* Corresponding author.

E-mail address: sean.benson@cern.ch (S. Benson).

the data are sent to permanent storage. The event reconstruction in the EFF is denoted as the *online reconstruction*.

In Run-I (2010–2012) of the LHC, the LHCb experiment used a processing model in which all events accepted by the HLT were sent to permanent offline storage containing all raw information from the detector. An additional event reconstruction performed on the LHC Computing Grid [3], denoted as the *offline reconstruction*, recreates particles in the event from the raw data using an improved detector calibration.

The upgrade of the computing infrastructure during the first long shutdown of the LHC (2013–2014), combined with efficient use of the EFF storage, provides resources for an online reconstruction in LHC Run-II (2015–2018) with a similar quality to that of the offline reconstruction. This is achieved through real-time automated calculation of the final calibrations of the sub-detectors.

With fully calibrated detector information available at the HLT level, it is possible to perform physics analyses with the information calculated by the HLT event reconstruction. In the *Turbo stream*, a compact event record is written directly from the trigger and is prepared for physics analysis by the Tesla application, which is named following the LHCb renowned physicist convention. This bypasses the offline reconstruction. The performance of a final analysis quality event reconstruction already in real time as the data arrive has the power to transform the experimental approach to processing large quantities of data.

The data acquisition framework is described in Section 2. An overview of the upgrades to the trigger and calibration framework in Run-II is provided in Section 3. The implementation of the Turbo stream including that of the Tesla application is described in Section 4, followed by the future prospects of the data model in Section 5.

2. The LHCb detector, data acquisition, and trigger strategy

The LHCb detector is a forward arm spectrometer designed to measure the properties of the decays of b -hadrons with high precision [4]. Such decays are predominantly produced at small angles with respect to the proton beam axis [5]. This precision is obtained with an advanced tracking system consisting of a silicon vertex detector surrounding the interaction region (VELO), a silicon strip detector located upstream of the dipole magnet (TT), and three tracking stations downstream of the magnet, which consist of silicon strip detectors in the high intensity region close to the beamline (IT) and a straw-tube tracker in the regions further from the beamline (OT). Neutral particles are identified with a calorimeter system consisting of a scintillating pad detector (SPD), an electromagnetic calorimeter preceded by a pre-shower detector (ECAL, PS), and a hadronic calorimeter (HCAL). Particle identification is provided by combining information from the ring-imaging Cherenkov detectors (RICH1 and RICH2), the wire chambers used to detect muons, and the calorimeter system.

2.1. Data readout and hardware trigger

Most interesting physics processes occur at event rates of less than 10 Hz in Run-I conditions, which can be compared to the 30 MHz at which bunches of protons are brought to collision. Reducing the output rate through the use of a trigger system is essential to reject uninteresting collisions, thereby using computing resources more efficiently. The detector front-end (FE) electronic boards connect to a set of common readout boards (RB) that limit the event output rate to 1.1 MHz. At LHCb a three-level trigger system is used consisting of a hardware level followed by two software levels. The level-0 hardware trigger (LO) reduces the input collision rate of ~ 30 MHz to the maximum output rate

allowed by the front-end electronics. The LO decision is based on algorithms running on field-programmable gate arrays (FPGAs), which achieve the necessary reduction in rate within a fixed latency of 4 μ s. Information from the ECAL, HCAL, and muon stations is used in FPGA calculations in separate LO algorithms. Decisions from these different hardware triggers are combined and passed to the readout supervisor (RS). The readout boards perform zero-suppression and interface the custom electronics to the readout network via Gigabit Ethernet links. The RS decides where in the EFF to send the event based on the state of the buffers and the available resources in the EFF. The EFF node address information is sent to the RB via optical links that also keeps them synchronised. Event raw data are distributed among the individual servers of the EFF using a simple credit-request scheme [6].

2.2. High level trigger

The LHCb high level trigger is a software application, executed on the EFF, that is implemented in the same Gaudi framework [7] as the software used for the offline reconstruction. This permits the incorporation of offline reconstruction software in the trigger, provided that it is sufficiently optimised to be used in real time. While no sub-detector upgrade took place in the first long shutdown of the LHC (2013–2014), the EFF was improved. The EFF now consists of approximately 1800 nodes, with 1000 containing 2 TB of hard disk space each and 800 nodes containing 4 TB each, giving a total of 5.2 PB. Each server node in the EFF contains 12/16 physical processor cores and 24–32 logical cores.

The first level of the software trigger (HLT1) reconstructs charged particle trajectories using information from the VELO and tracking stations. If at least one track is found that satisfies strict quality and transverse momentum criteria, then the event is passed to the second level of the software trigger (di-muon combinations may also trigger the first software level). The output rate of HLT1 is ~ 150 kHz.

The second level of the software trigger (HLT2) can use information from all sub-detectors to decide whether or not to keep an event for analysis and permanent storage. A full event reconstruction is performed such that HLT2 is then able to identify the most interesting events, with a final output rate of ~ 10 kHz.

3. Run-II data taking

During Run-I data taking, a buffer was created between the hardware trigger and the first software trigger level, deferring 20% of the events passing the hardware trigger and thereby utilising the EFF when the LHC was not providing proton collisions. The replacement of this buffer with the one between the two software levels introduces a complete split between an initial stage processing events directly from the LO (HLT1) and an asynchronous stage (HLT2), ensuring the EFF is used optimally.

From 2015 onwards, the two HLT software levels run independently, giving rise to the Run-II data flow depicted in Fig. 1. In order to process two independent trigger stages, a substantial modification to the trigger software was required (a detailed description is provided in Ref. [6]). This included the creation of an additional buffer between the two software levels. The flexibility in trigger processing that is provided by this buffer system allows the execution of high-quality alignment and calibration between HLT1 and HLT2.

The alignment and calibrations are described in detail Section 3.1. The different data streams created from the HLT2 selected events are described in Section 3.2. Among these is the Turbo stream in which information from the online reconstruction is persisted, detailed in Section 4.

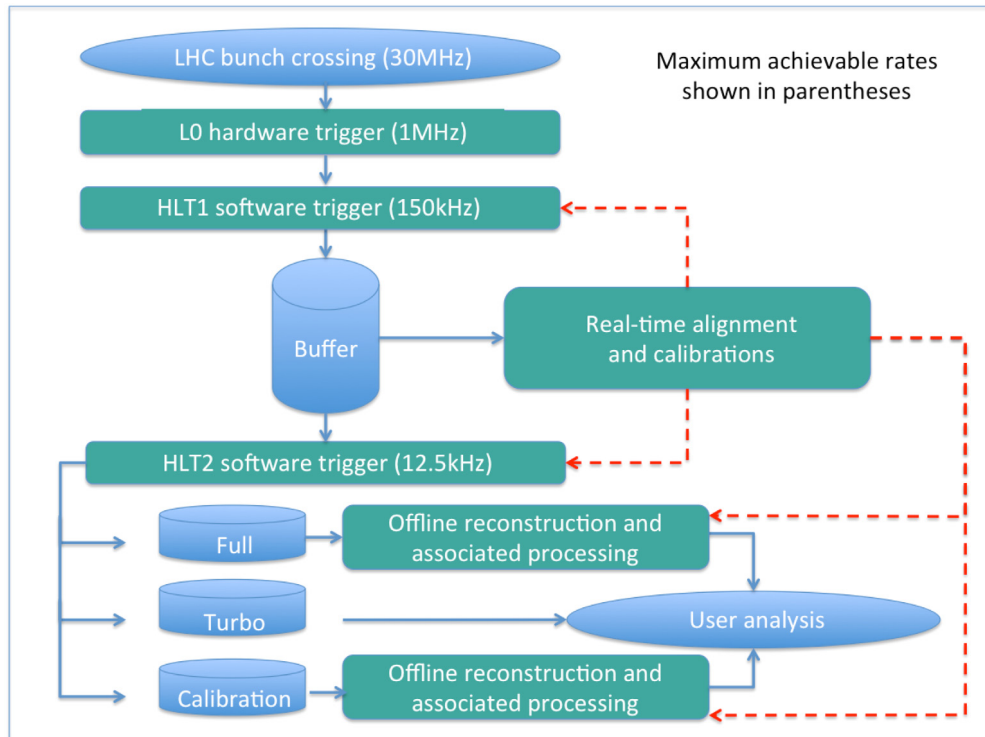


Fig. 1. Schematic diagram showing the overall data processing model in Run-II, where the blue solid line represents data flow, and the red dashed line the propagation of calibrations.

3.1. Real-time alignment and calibration

As described in Section 2.2, increased computing resources in the EFF allow for automated alignment and calibration tasks to supply high-quality information to the trigger software. This removes the need for further reprocessing.

In order to align and calibrate the detector, dedicated samples from HLT1 are taken as input. These calibrations are implemented before HLT2 has processed the data, which is a prerequisite for the successful functioning of the Turbo stream as it relies on information calculated by HLT2. The calibrations are also used by HLT1 for subsequent processing. The alignment and calibration tasks are performed at regular intervals. These intervals can be as frequent as each time a new LHC proton beam is formed or less frequent depending on the calibrations being calculated. The calibration tasks are performed in a few minutes using the nodes from the EFF. The resulting alignment or calibration parameters are updated if they differ significantly from the values used at a given time and are stored in a database as a function of time.

Full details of the real-time alignment procedure are provided in Ref. [8]. The complete calibration of the full detector is a complex enterprise. The achievement of automating and providing accurate calibrations within a few minutes is a substantial achievement without which analysis-quality reconstruction in HLT2 would be impossible.

3.2. Streaming and data flow

Data streams at LHCb are controlled through the assignment of routing bits to each event by the trigger software. Routing bits are set according to the trigger lines that select the event. A filter is applied based on the routing bit that allows for different streams to be sent to different places in permanent storage, as depicted in Fig. 1. These data are processed with the DIRAC software framework [3], which is used to manage all LHCb data processing on the Grid. The physics data streams in Run-II are the Full stream,

the Turbo stream, and the Calibration stream. Events are allowed to be present in more than one stream.

Events that are sent to the Full stream have luminosity information calculated, which is then stored in the data file as a file summary record (FSR). They undergo a further offline reconstruction using the sub-detector raw data banks, which contain the detector information. Subsequent analysis selections are applied, identifying decay channels of interest. After this processing is completed, files are merged to improve network utilisation in the LHC grid computing infrastructure, and the data are then available for physics analyses.

The events directed to the Turbo stream consist of the flagged physics lines and those triggered for luminosity accounting. The Turbo stream does not require further event reconstruction, since datasets are ready for user analysis directly after the generation of luminosity file summary records and the restoration of the trigger objects. As for the Full stream the resulting files are merged. This is illustrated in Fig. 2.

Calibration

Calibration candidates provide pure data samples of pions, kaons, protons, muons and electrons that can be used to determine efficiencies in a data-driven method. Most LHCb analyses apply selection requirements for variables that show some disagreement between real data and simulated data, for example Particle Identification. The determination of efficiencies using data is therefore preferred where possible.

Exclusive selections applied in the trigger allow candidates of highly abundant unstable particles, such as $D^{*+} \rightarrow D^0\pi^+$ to be directed to a dedicated stream. Events sent to the Calibration stream contain the stored trigger candidates in addition to the raw sub-detector data. Thus, both the offline reconstruction of the Full stream and the trigger restoration of the Turbo stream can be applied to Calibration stream events. The workflow of the Calibration stream is depicted in Fig. 2. In this way, the same particle decay candidates can be used to provide data driven corrections for both the online and offline event reconstructions.

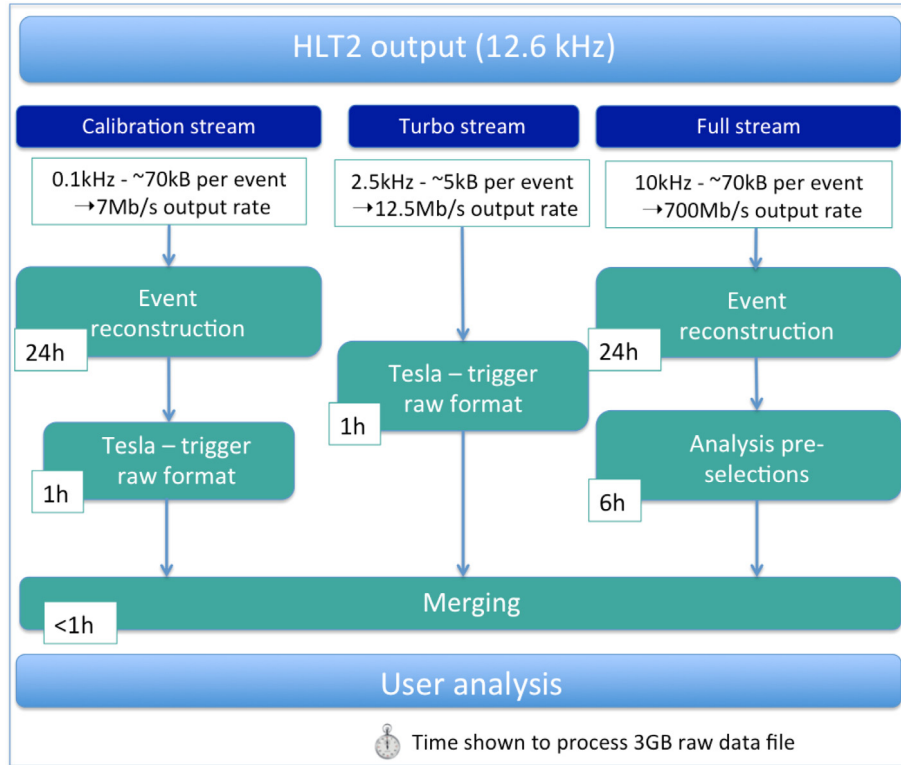


Fig. 2. Turbo data processing versus the traditional approach, as described in Section 4. The time taken for each step in hours is provided for a 3 GB raw data file. In addition, a calibration stream separates events for further processing to calculate data-driven efficiencies for both the Full and Turbo streams.

4. Implementation of the turbo stream

The concept of the Turbo stream is to provide a framework by which physics analyses can be performed using the online reconstruction directly. The schematic data flow of the Turbo stream compared to the traditional data flow (represented by the Full stream) is depicted in Fig. 2.

In the traditional data flow, raw event data undergoes a complete reconstruction taking 24 h for 3 GB of input data on a typical batch node. This additional reconstruction was designed for a data processing model in which final calibrations performed at the end of a data taking year were a significant improvement compared to the calibrations initially available. This is no longer needed since high quality calibrations are produced in real time. After the offline reconstruction, selection criteria based on typical b -hadron and c -hadron topologies are applied that identify the decay candidates for user analysis, taking an additional 6 h for the 3 GB of raw data. After a final merging step of less than one hour, the datasets are ready for user analysis. The approach taken by the Turbo stream is to save the particle candidates reconstructed in the HLT (equivalent to those produced after the selection stage in the traditional approach) inside the raw event. The time taken for the Tesla application to format the data in preparation for user analysis is approximately 1 h.

A clear advantage of the Turbo stream is that the event size is an order of magnitude smaller than that of the Full stream as all sub-detector information may be discarded. For standard Run-II conditions ~20% of the HLT2 selected events will be sent to the Turbo stream at a cost of less than 2% of the output bandwidth.

In order to perform physics analyses with the online reconstruction, decay candidates must appear in the same format as expected by the output of the traditional processing, such that the existing analysis infrastructure can be used. This is the purpose of the Tesla application. The high-level functions of the Tesla application are detailed in Section 4.1. The low-level design is described in Section 4.2.

4.1. The Tesla application

For events in the Turbo stream, the reconstructed decay candidates are stored in the raw data in the format usually reserved for detector level information. The Tesla application is subsequently used to perform multiple tasks on events sent to the Turbo stream.

The purpose of Tesla is to ensure that the resulting output is in a format that is ready for analysis. This means that the Tesla application must do the following:

- Compute the information that is necessary for the luminosity determination and store this in the output file, as described in Section 3.2.
- Place the HLT candidate decays in the output file in such a way that existing analysis tools function correctly with minimal modifications.
- Ensure that additional information calculated in the online reconstruction is accessible to standard analysis tools, for example event-level sub-detector occupancies and information calculated using the whole reconstructed collision.
- Discard the raw data corresponding to the sub-detectors, leaving only the information on the requested HLT candidates, trigger decisions and headers required for subsequent analysis. It should be noted that this requirement is only relevant for 2015 commissioning and will take place inside the trigger itself from 2016 onwards.

The Tesla application must be capable of processing simulated data sets. Thus, it must be able to associate reconstructed tracks and calorimeter clusters to the simulated particles that produced these signatures in the simulated detector. There should be also an option to protect the detector raw data, such that the offline reconstruction in the traditional data flow can coexist in simulated events alongside the physics analysis objects from the online reconstruction. The processing sequence used by the Tesla

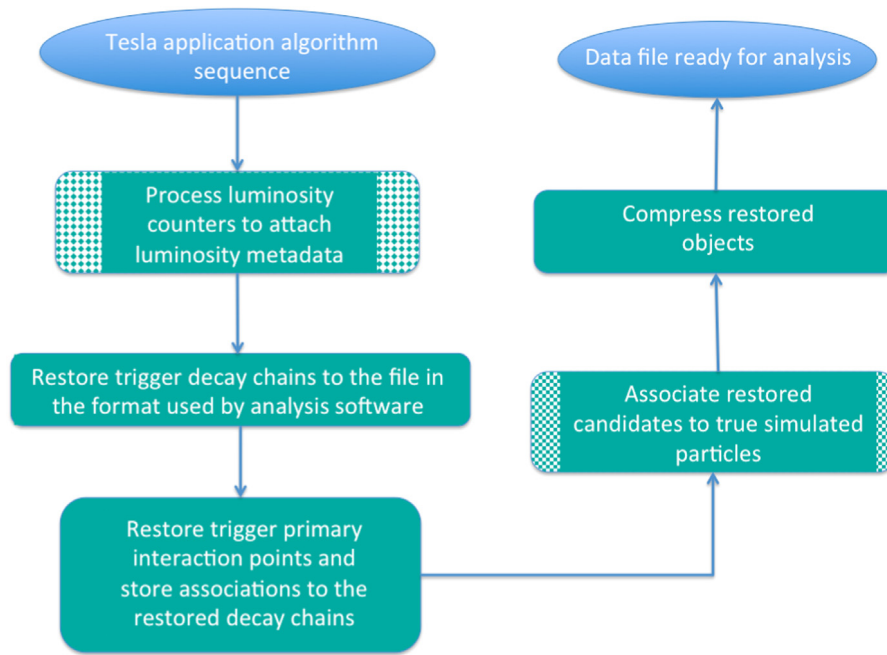


Fig. 3. The processing sequence used by the Tesla application. The hatched diamonds indicate a step is only applied to real data, whereas hatched squares indicate a step is only applied to simulated data.

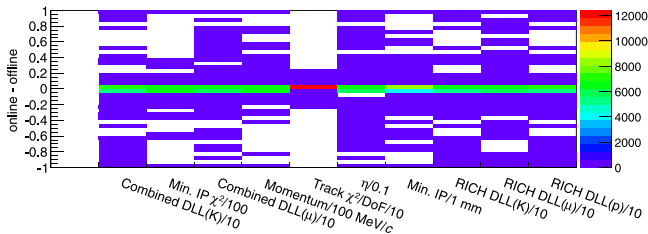


Fig. 4. Comparison between variables calculated by the online and offline event reconstructions for the pion in $D^0 \rightarrow K^- \pi^+$ decays of a single hour of LHCb data taking, where the colour represents the number of pion candidates. The variables are defined in the text.

application is depicted in Fig. 3. It should be noted that the objects created by the Tesla application are stored in different addresses in the output file than those used by the traditional processing so that there can be no interference. This is a requirement for the calibration stream as described in Section 3.2.

Monitoring and validation

In addition to providing analysts with data for physics measurements, the Tesla application allows for a new method with which to monitor the stability of the detector and the reconstruction. Monitoring is traditionally performed at LHCb by studying abundant decays such as $D^0 \rightarrow K^- \pi^+$ using a separate implementation of the offline reconstruction. The data used is that of a single run, which is 1 h in duration at most. The creation of histograms directly from the Tesla application permits the simultaneous analysis of several runs. The additional monitoring allows for variables calculated in the online and the offline event reconstructions to be compared to ensure consistency. This also serves the purpose of validating the use of the real-time alignment inside the offline processing. An example comparison is given in Fig. 4 of the reconstruction variables calculated for the pion in $D^0 \rightarrow K^- \pi^+$ decays. The variables shown are used commonly in physics measurements to isolate interesting decays and are defined as follows: the difference in log-likelihood between a given particle type and that of a pion both combining all sub-detector information (Combined DLL) and restricting to the RICH

sub-detector information (RICH DLL), the minimum difference in χ^2 of the primary vertex between the fits with and without the pion candidate (Min. IP χ^2), the track fit quality (Track χ^2/DoF), the reconstructed pseudorapidity (η) and momentum, together with the minimum impact parameter (Min. IP) with respect to the primary vertices. Comparing the online and offline calculated variables of two decay candidates requires that they are associated to the same detector hits thereby ensuring the same candidate is found.

4.2. Persistence of trigger objects

Measurements of CP -violating asymmetries along with searches for new states and rare decays are made using datasets that are produced from physics object classes, written in C++. These are stored in the hashed containers of the data summary tape (DST) format. In the Full stream, the size of the raw event is on average 70 kB. When the sub-detector banks are removed from the raw event, and replaced with the specific trigger candidates firing the Turbo stream triggers, the event size is decreased to around 5 kB.

During Run I data taking (2009–2012), some limited information from physics objects made inside the trigger was already placed inside the raw event. This allows the possibility to determine trigger efficiencies directly from the data using the methods described in Ref. [1]. The so-called selection reports allow for a C++ class, known as an HLT object summary, to save the members of any C++ physics object class in a key–value pair and point to other object summary instances. After the physics objects have been converted into object summaries, a dedicated algorithm writes them into the dedicated sub-bank of the raw data bank associated to the HLT. This is depicted in Fig. 5.

A complete physics measurement requires much more information than was saved in the selection reports in Run I. This means that many more classes must be placed into the raw event and more information must be saved about each individual class. In order to save entire decay chains, a novel pattern is required to describe the topology. This, combined with the information inside the summaries, allows for all the required information of the decay

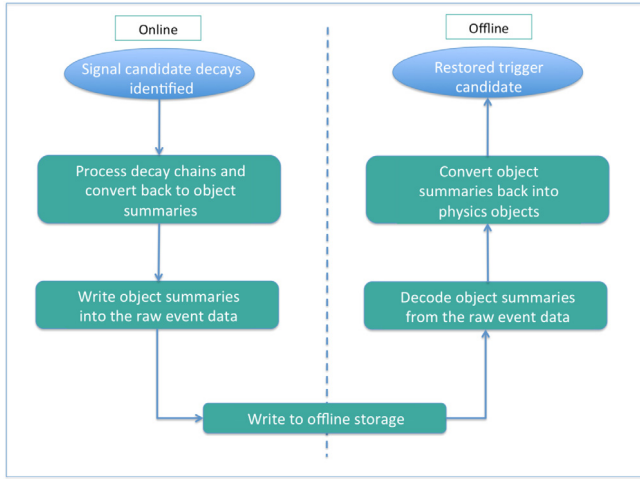


Fig. 5. Steps required to save and restore trigger objects.

to be saved. The pattern of the reports used for saving the decay topology is shown in Fig. 6.

One of the main advantages of the method is its flexibility as only relevant information contained in the object classes is persisted. Additional members added at a later date can also be stored. In order to preserve backwards compatibility, the version of the raw bank is written to its header. The version indicates to the dedicated converter tool exactly which piece of information about the class is present for a given location in the raw data stream. It is important that the restored data match the original data. In order to achieve this, the same dedicated converter tool is used both to read and to write the object summary. For a given analysis class, the position in the raw data stream is dedicated to a specific quantity. Individual static maps for each analysis class, such as particles and tracks, are used by the converter tool to keep track of exactly where each piece of information is stored. The raw bank version is included on an event-by-event basis, which means that multiple versions can be processed simultaneously, avoiding the issue of backwards compatibility.

4.3. Additional analysis requirements

While physics measurements mostly rely on information about the signal candidate, the complete event is often exploited to

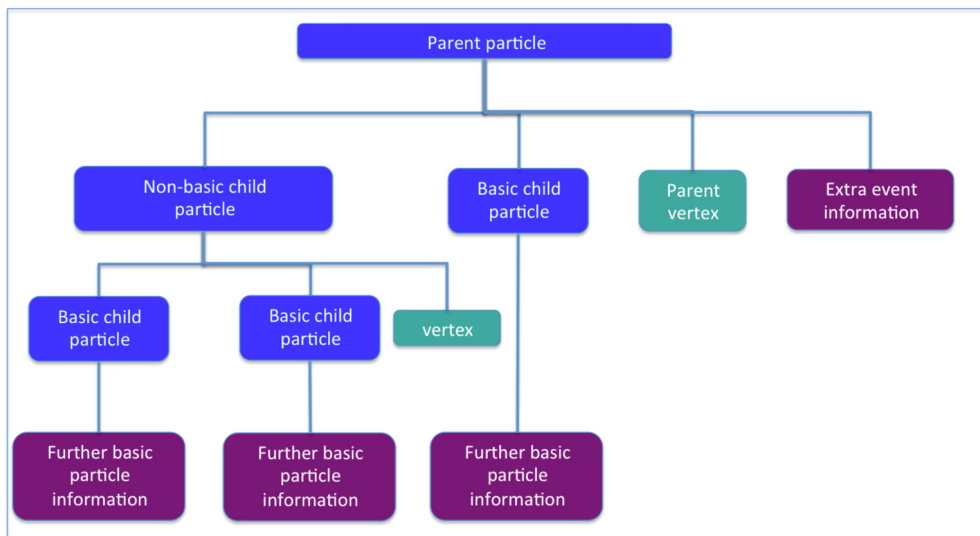


Fig. 6. Structure of stored objects. Each top-level object represents a parent particle, which then points to child particles along with associated information. Note that the example shown is for a two-level particle decay. The infrastructure allows for as many levels as needed.

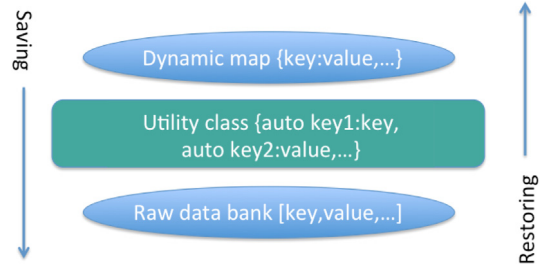


Fig. 7. Method used to save and restore dynamic maps.

calculate quantities that discriminate against background processes. A number of such quantities are employed in analyses, and the tools that calculate these quantities typically have parameters that are tailored to the needs of individual measurements. Examples of such quantities are isolation variables, which are measures of particle multiplicity within a region of the detector around triggered candidates. The necessary size of the region is analysis dependent, and some measurements use a set of regions with different sizes. This is a different use case, in which a static map is no longer appropriate to save and restore such data. In order to deal with such cases, an automatically generated utility class key is created to store separately the map key and corresponding value to be saved. This ensures that the key and data of the original dynamic map are both written to the raw data stream, as shown in Fig. 7.

Extra objects needed for analysis, such as primary vertices that have been fitted without the signal tracks, are persisted one level down from the top of the decay chain. Since these objects represent static analysis level classes, the usual persistence approach is used rather than the dynamic map.

4.4. Analysis using the online reconstruction

The intended purpose of the Tesla application and the Turbo stream infrastructure is to ensure that more decays can be collected and therefore more precise measurements performed than would have been possible under the Run-I computing model. Example data distributions showing clean decay signals directly from the information of the online reconstruction are provided in Fig. 8. The limits on the output rate using the Full stream mean that only a subset of the high-rate particle decays would have been collected without the Turbo stream infrastructure.

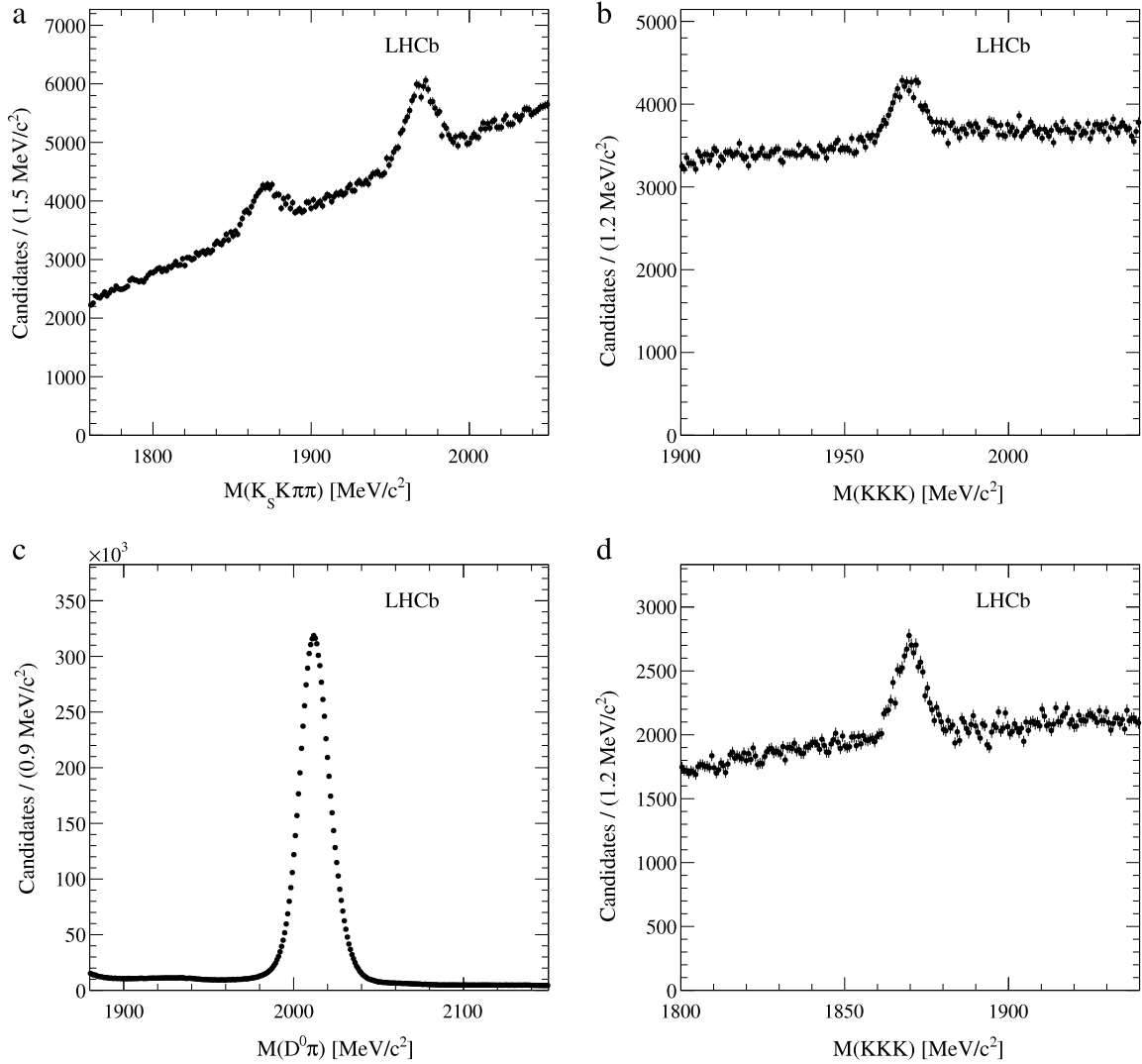


Fig. 8. Invariant mass distributions for candidates reconstructed in the $K_S^0 K^- \pi^+ \pi^+$ (a), $K^- K^+ K^+$ (b), $D^0 (\rightarrow K^- \pi^+) \pi^+$ (c), and $K^- K^+ K^+$ (d) state hypotheses, from 26.5 pb^{-1} of proton collision data taken in 2015. The peaks corresponding to D^+ , D_S^+ , and D^{*+} mesons can clearly be seen.

5. Outlook and future prospects

The use of the Turbo stream in 2015 proved to be successful. The first two published physics measurements from the LHCb experiment based on data collected in the 2015 run were based on the Turbo stream [9,10]. Around half of the HLT2 trigger lines currently persist the trigger reconstruction using the Turbo stream.

5.1. Use in the upgraded LHCb experiment

The upgrade of the LHCb experiment will see the luminosity rise from $4 \times 10^{32} \text{ cm}^{-2} \text{ s}^{-1}$ to $2 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$. In addition, the L0 hardware trigger will be removed. Therefore, the software trigger will need to cope with input rates that are ~ 30 times larger than the current trigger input rate. The output rate of b -hadrons, c -hadrons and light long-lived hadrons will increase significantly, as shown in Table 1. In order to accommodate the increase in rate, the use of specific selection triggers will become increasingly necessary. The Turbo model will become increasingly utilised as an increased retention will have a direct impact on physics reach. With the rate of interesting events being so high, the output bandwidth becomes a concern. The average size of a raw event in the LHCb upgrade is anticipated to be $\sim 100 \text{ kB}$. An event rate

Table 1

Rates of hadrons in Run I compared to the expected rate in the upgraded LHCb experiment [11], including expected trigger and reconstruction efficiencies.

Particle type	Run I (kHz)	Upgrade (kHz)
b -hadrons	17.3	270
c -hadrons	66.9	800
Light long-lived hadrons	22.8	264

of 20 kHz in traditional model will then be 2 GB/s. Assuming $1.5 \times 10^7 \text{ s}$ of data taking each year, storage space at the level of 30 PB/year would be required to save the data in raw form. Usually multiple copies of the raw data are stored to ensure safety against data loss. In addition, the size of the reconstructed event is larger than the raw form, meaning the total storage needs would be at the level of 300 PB/year. The traditional model therefore does not scale to the LHCb upgrade regime. The Turbo model would only require 100 MB/s and would provide $20\times$ the output event rate of the traditional approach with the same resources, assuming all events in the upgraded LHCb experiment use the Turbo model. A mixture of Turbo and traditional approaches is expected to be used as many analyses require the use of full event information.

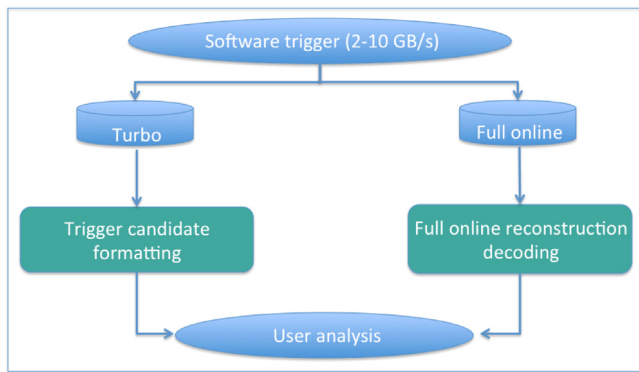


Fig. 9. Possible LHCb Upgrade data processing model.

5.2. Extended use of the trigger reconstruction

The methods described in Sections 3 and 4 use the trigger reconstruction to reduce the event size thus relaxing the upper limit on the event rate. The single reconstruction and the removal of a layer of selections also simplify analyses, potentially reducing many sources of systematic uncertainty. These benefits would also be seen if the complete reconstructed event in the trigger was available for analysis. The event model is depicted in Fig. 9. In a similar fashion to the Turbo model, where analysis classes for candidates are saved inside the raw data banks, the physics objects created by the online reconstruction may also be saved to the raw data bank. With a formatting level to prepare the data for user level analysis tools, an additional reconstruction can be avoided therefore reducing the computing time. With appropriate filters inside the trigger, the decision of whether to keep the raw event or not could be made on an event-by-event basis.

6. Summary

The Tesla toolkit allows for analyses to be based on the event reconstruction that is performed by the LHCb HLT. By design, the Tesla output files are compatible with existing analysis framework software with minimal changes required from analysts.

The event reconstruction performed by the HLT is of sufficient quality for use in physics analyses because the detector is aligned and calibrated in real time during data taking. This is in turn made possible through the upgraded computing infrastructure introduced in the first long shutdown of the LHC and the decision to buffer data from the first software trigger level.

The successful commissioning of this concept in 2015 has allowed multiple analyses to be performed based on this model. The higher output rates of the LHCb upgrade will make the approach increasingly necessary.

Acknowledgements

We thank the technical and administrative staff at the LHCb institutes. We acknowledge support from CERN and from the national agencies: CAPES, CNPq, FAPERJ and FINEP (Brazil); CNRS/IN2P3 (France); BMBF, DFG and MPG (Germany); INFN (Italy); FOM and NWO (The Netherlands); SNSF and SER (Switzerland); STFC (United Kingdom). We acknowledge the computing resources that are provided by CERN, IN2P3 (France), KIT and DESY (Germany), INFN (Italy), SURF (The Netherlands), PIC (Spain), GridPP (United Kingdom), RRCKI and Yandex LLC (Russia), CSCS (Switzerland), IFIN-HH (Romania), CBPF (Brazil), PL-GRID (Poland) and OSC (USA). We are indebted to the communities behind the multiple open source software packages on which we depend.

References

- [1] R. Aaij, et al., JINST 8 (2013) P04022. [arXiv:1211.3055](https://arxiv.org/abs/1211.3055).
- [2] LHCb collaboration, R. Aaij, et al., Int. J. Mod. Phys. A 30 (2015) 1530022. [arXiv:1412.6352](https://arxiv.org/abs/1412.6352).
- [3] F. Stagni, P. Charpentier, J. Phys. Conf. Ser. 368 (2012) 012010.
- [4] LHCb collaboration, A.A. Alves Jr., et al., JINST 3 (2008) S08005.
- [5] LHCb collaboration, R. Aaij, et al., Phys. Lett. B 694 (2010) 209. CERN-PH-EP-2010-029, LHCb-PAPER-2010-002, [arXiv:1009.2731](https://arxiv.org/abs/1009.2731).
- [6] M. Frank, et al., J. Phys. Conf. Ser. 513 (2014) 012006.
- [7] G. Barrand, et al., Comput. Phys. Comm. 140 (2001) 45.
- [8] G. Dujany, B. Storaci, J. Phys. Conf. Ser. 664 (8) (2015) 082010.
- [9] LHCb collaboration, R. Aaij, et al., J. High Energy Phys. 10 (2015) 172. LHCb-PAPER-2015-037, CERN-PH-EP-2015-222, [arXiv:1509.00771](https://arxiv.org/abs/1509.00771).
- [10] LHCb collaboration, R. Aaij, et al., J. High Energy Phys. (2016) [http://dx.doi.org/10.1007/JHEP03\(2016\)159](https://doi.org/10.1007/JHEP03(2016)159), [arXiv:1510.01707](https://arxiv.org/abs/1510.01707), LHCb-PAPER-2015-041, CERN-PH-EP-2015-272, [arXiv:1510.01707](https://arxiv.org/abs/1510.01707).
- [11] C. Fitzpatrick, V. Gligorov, Anatomy of an upgrade event in the upgrade era, and implications for the LHCb trigger, LHCb-PUB-2014-027, 2014.