Journal of Physics: Conference Series **664** (2015) 082004 doi:10.1088/1742-6596/664/8/082004

The LHCb Turbo Stream

Sean Benson[1], Vladimir Gligorov[1], Mika Anton Vesterinen[2], John Michael Williams[3]

[1]CERN, Geneva, Switzerland,
[2]Physikalisches Institut, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany,
[3]Massachusetts Institute of Technology, Cambridge, MA, United States.

E-mail: sean.benson@cern.ch

Abstract. The LHCb experiment will record an unprecedented dataset of beauty and charm hadron decays during Run II of the LHC, set to take place between 2015 and 2018. A key computing challenge is to store and process these datasets, which will limit the maximum output rate of the LHCb trigger. So far, LHCb has written out a few kHz of events containing the full raw sub-detector data, which are passed through a full offline event reconstruction before being considered for physics analysis. Charm physics in particular is limited by trigger output rate constraints. A new streaming strategy includes the possibility to perform the physics analysis with candidates reconstructed in the trigger, thus bypassing the offline reconstruction and discarding the raw event. In the Turbo stream the trigger will write out a compact summary of physics objects containing all information necessary for analyses, and this will allow an increased output rate and thus higher average efficiencies and smaller selection biases. This idea will be commissioned and developed during 2015 with a selection of physics analyses. It is anticipated that the turbo stream will be adopted by an increasing number of analyses during the remainder of LHC Run II (2015-2018) and ultimately in Run III (starting in 2020) with the upgraded LHCb detector.

1. Introduction to LHCb

The LHCb detector is a forward arm spectrometer, specialising in the measurement of the properties of long-lived charged particles originating from the decays of B mesons with high precision [1]. 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 occupancy 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 with a pre-shower detector placed in front (ECAL, PS), and a hadronic calorimeter (HCAL). Particle identification is provided by the ring-imaging Cherenkov detectors (RICH1 and RICH2), the multi-wire proportional chambers used to detect muons, and also the calorimeter system.

2. Run I data taking

Large backgrounds are present at hadron colliders. At the nominal LHCb luminosity of $4 \times 10^{32}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, over 200k $b\bar{b}$ pairs are produced every second. The charm cross-section is

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

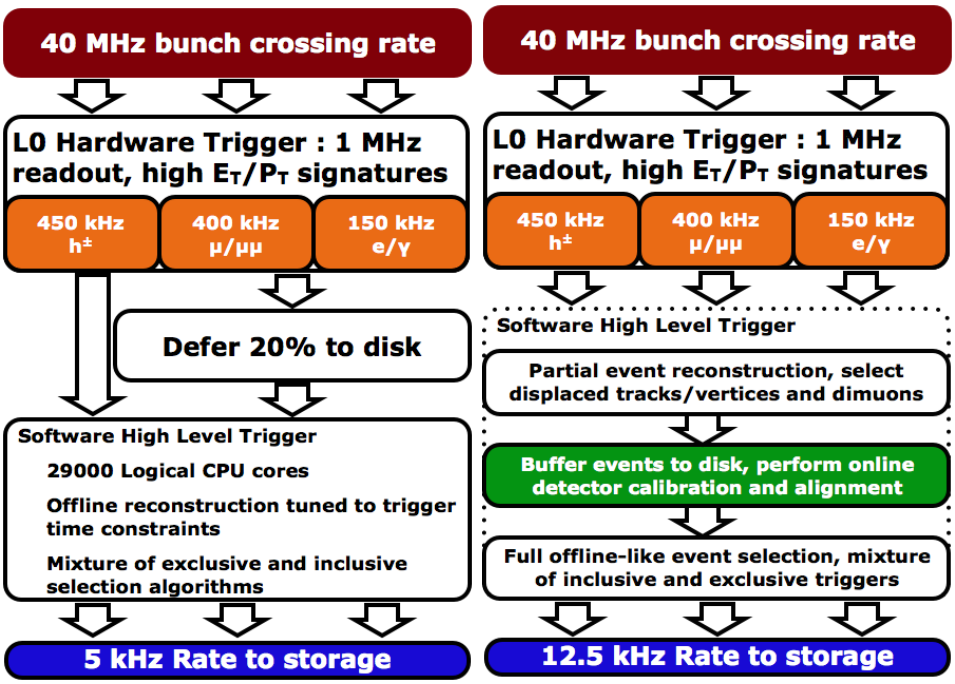

Figure 1. Schematic diagram of the LHCb trigger data-flow in Run I data taking (left) compared to the proposed data-flow in Run II (right).

a factor of 20 larger than the corresponding beauty cross-section. This means that reducing the output rate through the use of a trigger becomes essential not only for background rejection, but also to distinguish between different signals. At LHCb a three-level trigger is used. The level-0 hardware trigger initially reduces the rate to 1 MHz from the input collision rate of at most 30 MHz (limited by the bunch crossing rate of 40 MHz and taking account of the necessary gaps in the bunch trains). In Run I, events which were selected by the hardware trigger were passed through two software levels, giving a final output rate of 5 kHz.

2.1. Data readout and hardware trigger

The maximum rate of events that can be read out of the detector is imposed by the front-end (FE) electronics and corresponds to a rate of 1.1 MHz. In order to determine which events are kept, hardware triggers based on field-programmable gate arrays (FPGAs) are used with a maximum latency of 4 μs. Information from the ECAL, HCAL, and muon stations is used in FPGA calculations in separate L0 triggers. Decisions from the different hardware triggers are combined and passed to the readout supervisor board (RS). The front-end electronics connect to the 320 RS boards via 5000 optical links that allow for a through-put of 4 TBs^{-1} [2]. The RS boards perform zero-suppression and interface the custom electronics to the readout network via Gigabit Ethernet. Each readout supervisor board holds a fragment of an event. The average event fragment size is 120 bytes. An IP/Ethernet overhead of 58 bytes must be accounted for so fragments are packed into multi-event packets (MEP) to improve network utilisation. The MEP packing factor is around 10 on average. In practice this varies as the RS controls the packing factor dynamically, for example to avoid MEPs with mixed trigger types. The RS is capable of emulating the state of the FE buffers to protect against overflow. Based on the state of the buffers and the available resources in the event filter farm (EFF), the RS can decide whether or not to pass the event.

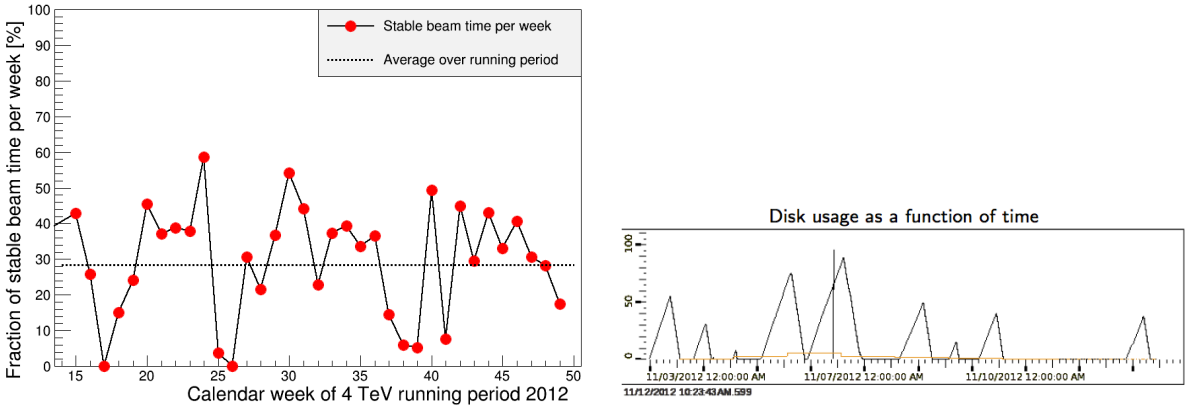

Figure 2. Fraction of time spent in stable beams as a function of calendar week [4] (left) and EFF disk usage as a function of time (right).

2.2. High level trigger

The LHCb high level trigger (HLT) is a C++ application, executed on the EFF, that is written in the same Gaudi framework as the offline software. This then allows for offline software to be easily incorporated into the trigger, providing the software is optimised sufficiently for online resources. In Run I, the EFF consisted of approximately 1500 dual processor units, divided into 50 sub-farms. The processors used in the EFF contain between 8 and 16 cores each. This allowed for 26110 simultaneous instances of the HLT (making use of hyper-threading technology). The destination in the EFF for a given MEP is chosen based on a credit scheme. The credit associated with each destination functions as a counting semaphore. The nodes in the farm must declare themselves as ready to receive at the beginning of a run. At the end of the processing, MEP requests must be submitted to the RS to obtain more. A schematic diagram showing the trigger data flow in Run I is depicted in Figure 1. In 2011, the L0 output rate was 870 kHz [3]. With the resources available in the EFF, this provided approximately 30 ms for HLT processing.

The first level of the software trigger performs a track fit using information from the VELO and the tracking stations. If a single, high quality track is found based on transverse momentum (p_T) and track fit quality criteria, this is passed to the second level of the software trigger (di-muon combinations may also trigger the first software level).

The second level of the software trigger is capable of using information from all sub-detectors to decide whether or not to keep an event. A full event reconstruction is performed. This allows for inclusive and exclusive final states to trigger the event. Around 40 % of the trigger output rate is dedicated to inclusive topological trigger lines, another 40 % is dedicated to exclusive charm triggers, with the rest divided among di-muon lines.

2.3. Trigger deferral

On average the time spent in stable running for the LHC during 2012 was around 30 %. This percentage is typical of a nominal data taking year due to planned technical stops and machine development phases and the ramping of the LHC dipole magnets between data taking fills. In addition, unplanned maintenance is often required. This means that if no data is buffered, the EFF would be active 30 % of the time and idle 70 % of the time. The fraction of time spent in stable beams in addition to the disk usage over time is shown in Figure 2. In 2012, farm nodes were equipped with 1 TB of local storage space, yielding a total data buffer of 1.5 PB.

During 2012 data taking, 25 % of the L0 output was buffered into the worker nodes of the HLT EFF [4] in order to keep the EFF active during LHC downtime.

3. Run II data taking

Due to the success of deferring the high level trigger processing, the decision was taken to expand the deferral. To better make use of the deferral system, the trigger is split into two independent programs, with buffering taking place after the first level. This then allows all data passing the first software level to be placed into the buffer (rather than 20 % of the L0 output). Figure 1 shows the trigger data flow as anticipated in Run II.

While no sub-detector upgrade took place in the first long shutdown (LS1), the EFF did undergo an upgrade. This involved an upgrade of the local storage, to provide total buffer space of 5.2 PB. In addition, new farm nodes based on Intel E5-2630v3 processors have been installed. This means that with the total resources available in the EFF, there is enough space in the buffer for around 10 days of continuous data taking. Accounting for the LHC duty cycle of $\sim 30\,\%$, this leaves ~ 50 ms to evaluate the first software level and ~ 800 ms to evaluate the second level per event.

With the increased time allowed in the second trigger level, the reconstruction in the trigger can be brought into line with the quality achieved offline. In addition automated calibrations can provide the same quality of observables used to distinguish signal from background as achieved offline, described in detail in Section 3.1.

In order to process two independent trigger stages, a substantial modification to the trigger software was required (a detailed description is provided in Ref. [4]). In each level of the trigger, readout functions are separated into independent tasks running asynchronously, where readout functions can be the assembly of event fragments as sent by the front end electronics, or the sending of accepted events to long term storage. This means the basic building blocks for processing event data can be reused (consisting of a producer, buffer manager and consumer). The producer task is responsible for reading and assembling data from data sources. Once read and assembled, the data block is placed in a shared memory area. This means that the buffer manager (BM) and producer are ready to process the next read operation. Each time an event is declared to the BM, a consumer task is activated, which releases the memory after it has finished processing. The producer tasks and reading tasks can be executed asynchronously, provided enough space is available for read operations. The BM is managed by a separate task on each node. This creates and initialises the shared memory area, in addition to handling requests for producers and consumers. This buffer system, which has been created for Run II, provides the required flexibility to enable the second trigger software level to be ran independently of the first level. This therefore facilitates the implementation of the offline alignment and calibrations, described in Section 3.1, in addition to the implementation of the Turbo stream, detailed in Section 4.

Conditions in Run II and Run III are expected to be very challenging when compared to Run I. Indeed collisions in the conditions of Run III will contain a reconstructible b-hadron 2 % of the time and a reconstructible c-hadron 24 % of the time [5]. Higher output rates are then essential to maintain efficiencies as close to those obtained in Run I as is possible. The 3 kHz of Run I output rate will then rise to 12.5 kHz sent offline in Run II, which is bounded by the space budget available on tape. It is planned that in Run II 10 kHz will be saved and processed as performed in Run I, while the remaining 2.5 kHz will be processed through the Turbo stream.

3.1. Real-time alignment and calibration

As has been mentioned previously, increased computing power in the EFF allows for automated alignment and calibration tasks, giving offline quality information inside the trigger software. This then allows for no further reprocessing until the end of 2015 at the earliest. Full details of the real-time alignment procedure are provided in Ref. [6].

In order to align and calibrate the detector, dedicated samples from the first software trigger level are used. The alignment and calibrations are performed at regular intervals. These intervals

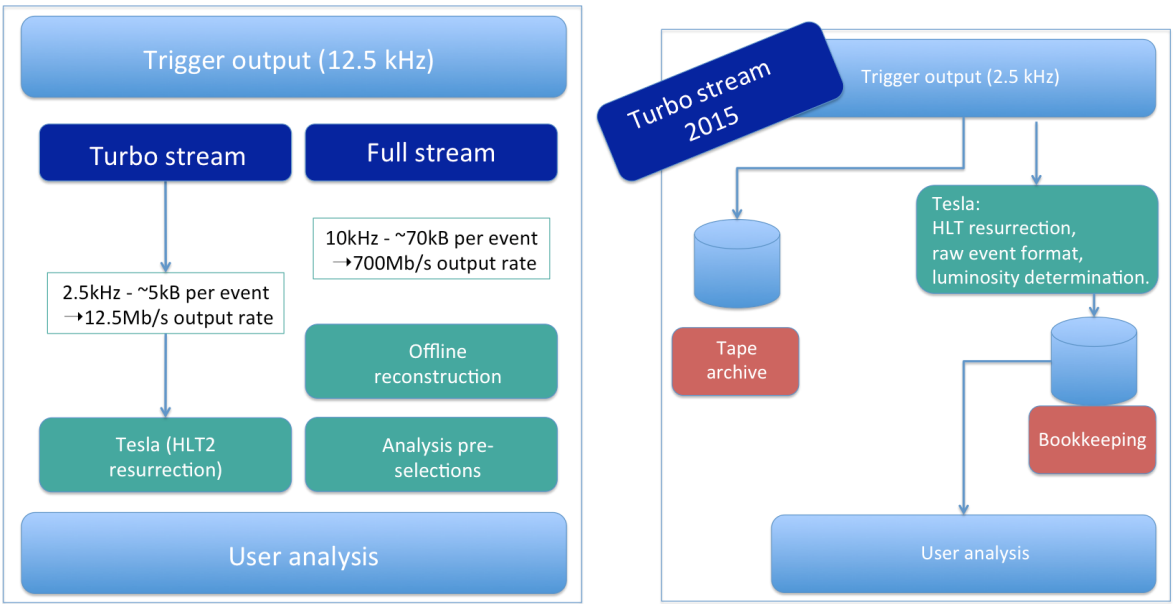

Figure 3. Turbo stream and Full stream processing stages (left), and data flow of the Turbo stream processing model in 2015 (right).

can either be at the beginning of the run, fill or less frequently depending on the task. The calibration tasks are performed in a few minutes using the nodes from the EFF. The resulting alignment or the calibration constants are updated if they differ significantly from the currently used values.

The major detector alignment and calibration tasks consist of:

- Alignment of the VELO and tracking stations.
- RICH mirror alignment.
- Global time alignment of the OT.
- RICH photon detector refractive index calibration.

4. The Turbo stream and its implementation

With offline quality information inside the trigger, the question arises as to whether or not an additional offline reconstruction is required. The concept of the Turbo stream is therefore to provide a framework by which a physics analysis can be performed using the trigger reconstruction directly. After commissioning of the concept, data events sent to the Turbo stream will persist only the candidates identified by the trigger reconstruction, discarding the rest of the event. The schematic data flow of the Turbo stream compared to the traditional data flow (denoted the Full stream) is depicted in Figure 3. One obvious advantage of the Turbo stream is the raw event size, which is an order of magnitude smaller than that of the Full stream, described in detail in Section 4.3. For Run II data taking, this results in 20 % of the trigger selections at a cost of less than 2 % of the output bandwidth.

The streaming framework required for the Turbo stream is described in Section 4.1; the Tesla application, responsible for the preparing Turbo stream events for the analysis framework, is described in detail in Section 4.2, with the technicalities of storing and resurrecting trigger objects detailed in Section 4.3.

4.1. Streaming and data flow

Data streams at LHCb are controlled through routing bits inside the trigger software. These are usually defined to record the firing of physics or luminosity trigger lines. In Run II, routing bits will also record the physics trigger lines flagged to go to the Turbo stream. The events directed to the Turbo stream will then consist of the aforementioned flagged physics lines and those triggered for luminosity accounting. Different streams are sent to different places in offline storage. These are then processed with the DIRAC software framework, used to manage all LHCb computing operations on the GRID [7].

Events sent to the Full stream have luminosity information calculated and stored as file summary records (FSRs). The events then undergo a further offline reconstruction, and consequently analysis pre-selections are applied, identifying decay channels, which can then be picked up after the data files have been merged. The total time for a 3 GB raw file to be processed through the entire chain is roughly 30 hours.

The Turbo stream does not require a further reconstruction. Turbo stream data is ready for dataset generation after luminosity file summary records have been created, the trigger objects resurrected, and the resulting data files merged.

For the purposes of commissioning the Turbo stream, events sent to the Turbo stream for measurements in early 2015 will also separately undergo offline reconstruction in order to validate and cross check the process.

4.2. The Tesla application

In 2015, the expected 2.5 kHz of the Turbo stream will contain the full raw event (to allow the previously mentioned validations to take place) with the enlarged reports containing all needed information on the decays from triggers sent to the Turbo stream.

The Tesla application is then used to perform multiple tasks such that the resulting output is in a format that is ready for analysis dataset production, and only contains a specific sub-set of the raw event that defines the Turbo stream. This means that the Tesla application must:

- Calculate the FSRs needed for luminosity determination and write this to the output file.
- Place physics objects in the output file in such a way that existing tools for dataset production work with minimal or ideally no modifications.
- Delete the sub-detector sub-banks, leaving only the enlarged reports, trigger decision banks and headers required for subsequent processing.

4.3. Storing and resurrecting trigger objects

Ultimately, it is the case that measurements of CP-violating asymmetries, branching fractions, and other observables for which LHCb was designed, are made using datasets that are produced from instances of physics object classes, written in C++, that are stored in the transient event store of the data summary tape (DST). The raw event persisted to offline storage was designed to primarily contain streams of sub-detector hits. 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 size decreases to the level of 5 kB.

During Run I data taking, some limited information from physics objects made inside the trigger was already placed inside the raw event. This infrastructure allowed for a data-driven determination of trigger efficiencies to be performed using the TISTOS method, explained in detail in Ref. [3]. The so-called selection reports allow for a utility C++ class, known as a 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 physics objects are converted to object

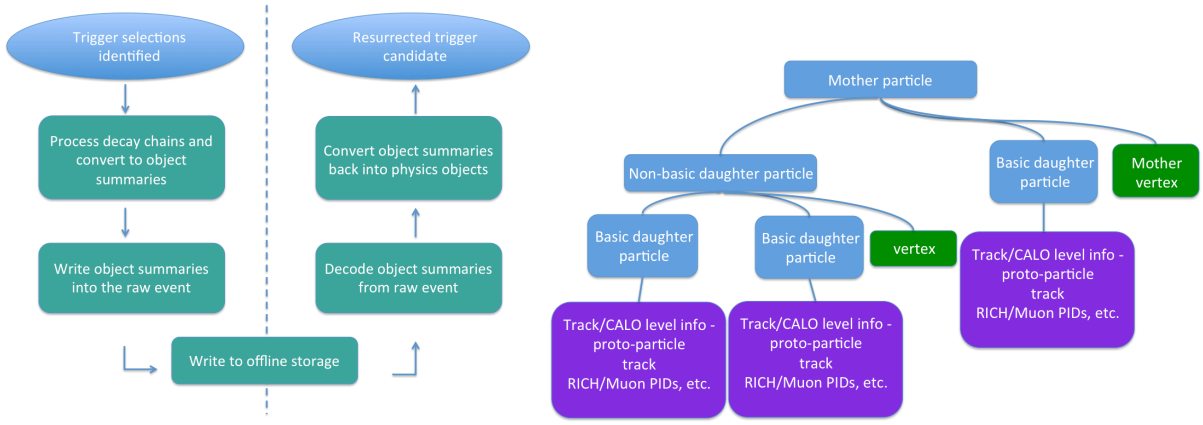

Figure 4. Schematic diagram showing the writing of trigger objects into the raw event along with the corresponding resurrection (left), and pattern of the reports used to identify decay topologies (right).

summaries, a dedicated writer streams these into the reports sub-bank of the raw event. This is depicted in Figure 4.

In order to accommodate all of the required information necessary to publish an analysis, much more information is required on the selected decay chains than was saved in Run I. This means that many more classes must be placed into the raw event. 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 on the decay to be saved. The pattern of the reports saving the decay topology is shown in Figure 4.

One of the main advantages of this method for saving the physics objects is flexibility. The persisted data can be chosen to save memory and avoid duplication. In order that the chosen information can be changed at will without backwards compatibility worries, the raw bank version is used. This is written in to the header of each bank. The version then indicates to the dedicated converter tool which map of key-value pairs is required for each object.

Clearly it is important that the resurrected data matches the data written in. In order to achieve this, the same dedicated converter tool is used to resurrect the object summary as that used to write it. For a given event, the raw bank version written to the raw event allows the converter tool to find the same map as that used to encode the data to the raw bank. This is included on an event-by-event basis and then means that multiple versions can be processed simultaneously, avoiding the issue of backwards compatibility.

5. Future plans

It is foreseen that as confidence is gained in performing analyses using data from the Turbo stream, more and more analyses will choose to move to the Turbo stream format. In particular charm analyses, which in many cases make use of exclusive triggers, would benefit as pre-scales could be avoided which may harm analyses such as charm spectroscopy.

The LHCb upgrade program, in which the RICH detectors and tracking stations will undergo improvements is expected to take place in long shutdown 2 (LS2) of the LHC schedule in 2018. After the LHCb upgrade program has been implemented, the majority of analyses will be sent to the Turbo stream. In the upgraded LHCb experiment, a $5\,\mathrm{GBs}^{-1}$ output bandwidth from the HLT to storage is foreseen. If all events were sent to the Turbo stream, this would allow for 1 MHz of events to be saved. The reality is that some analyses will always need the complete event and not specific candidates, therefore a mixture of the Full stream and Turbo stream format is a probable outcome.

6. Summary

Infrastructure has been created, by which physics analyses can be performed using physics objects created through the trigger reconstruction. This is made possible by the improvements incorporated in the trigger and data acquisition during the first long shutdown of the LHC. The improved computing power of the EFF combined with the strategy to buffer the data completely between the first and second software trigger levels allow for a much larger processing time budget in the second software trigger level. Together with automated alignment and calibrations performed in real-time, this allows for offline quality information to be calculated inside the trigger.

The use of the trigger reconstruction allows the raw event to decrease in size by an order of magnitude, allowing for 20 % of the trigger rate to equate to less than 2 % of the output bandwidth. In addition, analyses are simplified due to the removal of a separate reconstruction and additional selection level that would otherwise require accounting. However, this does come at the cost of not having the capacity to use an improved reconstruction offline, though automation of the alignment and calibrations means this is no longer necessary. The removal of a separate reconstruction and the corresponding pre-selection on that reconstruction also allows data to be ready in a small fraction of the previous time taken.

References

[1] LHCb Collaboration, A. Alves *et al.*, *The LHCb Detector at the LHC*, JINST **3** (2008) S08005.
[2] F. Alessio *et al.*, *The LHCb readout system and real-time event management*, IEEE Trans. Nucl. Sci. **57** (2010) 663.
[3] R. Aaij *et al.*, *The LHCb Trigger and its Performance in 2011*, JINST **8** (2013) P04022, `arXiv:1211.3055`.
[4] M. Frank, C. Gaspar, E. v. Herwijnen, B. Jost, and N. Neufeld, *Deferred High Level Trigger in LHCb: A Boost to CPU Resource Utilization*, J. Phys. Conf. Ser. **513** (2014) 012006.
[5] C. Fitzpatrick and V. Gligorov, *Anatomy of an upgrade event in the upgrade era, and implications for the LHCb trigger*, LHCb-PUB-2014-027 (2014).
[6] G. Dujany and B. Storaci, *Real-time alignment and calibration of the LHCb Detector in Run II*, J. Phys. Conf. Ser. (CHEP 2015) (2015).
[7] LHCb, F. Stagni and P. Charpentier, *The LHCb DIRAC-based production and data management operations systems*, J. Phys. Conf. Ser. **368** (2012) 012010.