The LHCb Data Acquisition and High Level Trigger Processing Architecture

M. Frank, C. Gaspar, B. Jost, N. Neufeld
CERN, 1211 Geneva 23, Switzerland
E-mail: `Markus.Frank@cern.ch`

Abstract. The LHCb experiment at the LHC accelerator at CERN collects collisions of particle bunches at 40 MHz. After a first level of hardware trigger with an output rate of 1 MHz, the physically interesting collisions are selected by running dedicated trigger algorithms in the High Level Trigger (HLT) computing farm. This farm consists of up to roughly 25000 CPU cores in roughly 1750 physical nodes each equipped with up to 4 TB local storage space. This work describes the LHCb online system with an emphasis on the developments implemented during the current long shutdown (LS1). We will elaborate the architecture to treble the available CPU power of the HLT farm and the technicalities to determine and verify precise calibration and alignment constants which are fed to the HLT event selection procedure. We will describe how the constants are fed into a two stage HLT event selection facility using extensively the local disk buffering capabilities on the worker nodes. With the installed disk buffers, the CPU resources can be used during periods of up to ten days without beams. These periods in the past accounted to more than 70 % of the total time.

1. Introduction

LHCb is a dedicated B-physics experiment at the LHC collider at CERN [1]. The LHCb detector was designed to record proton-proton collisions at a rate up to 40 MHz delivered by LHC at a center of mass energy up to 14 TeV. LHCb exploits the finite lifetime and large mass of charmed and beauty hadrons to distinguish heavy flavor particles from the background in inelastic pp scattering. The first level trigger (Level 0), which reduces the rate of accepted events to 1 MHz, is hardware based. The second level or High Level Trigger (HLT) is purely software based. As shown in Figure 1, the readout boards (Trigger ELectronics Level 1 board/TELL1) send data from particle collisions at a rate of roughly 1 MHz through a switching network to the HLT farm nodes, where dedicated algorithms compute the decision on whether the event is to be accepted. Events with a positive decision are sent to the storage system and subsequently to the LHCb computing grid portal [2] for later offline analysis. The HLT hardware consists of roughly 1750 dual processor sockets grouped into 56 sub-farms, which host about 50000 trigger processes. These processes execute on heterogeneous worker nodes with between 16 and 32 physical cores, each core equipped with roughly 2 GB of memory. Accepted events to be kept for later offline analysis are sent to the storage system, where the data streams get partially replicated and sent to the monitoring facilities, the Monitoring and the Reconstruction farm.
The experiment control system (ECS) handles the configuration, monitoring and operation of all equipment. All macroscopic entities of the ECS are modeled as Finite State Machine (FSM) entities [3], grouped together in a tree structure. The state of every higher level node summarizes the state of its children. The FSM tree mechanism is used to describe the functioning of the

processor farms such as the HLT processor farm, where at the lowest level processes are modeled as FSM elements, a set of processes is grouped into a node, a set of nodes describes a sub-farm and finally the set of sub-farms represents the high level trigger. All computing elements are shared and several concurrent instances may be mapped to the HLT farm or parts thereof.

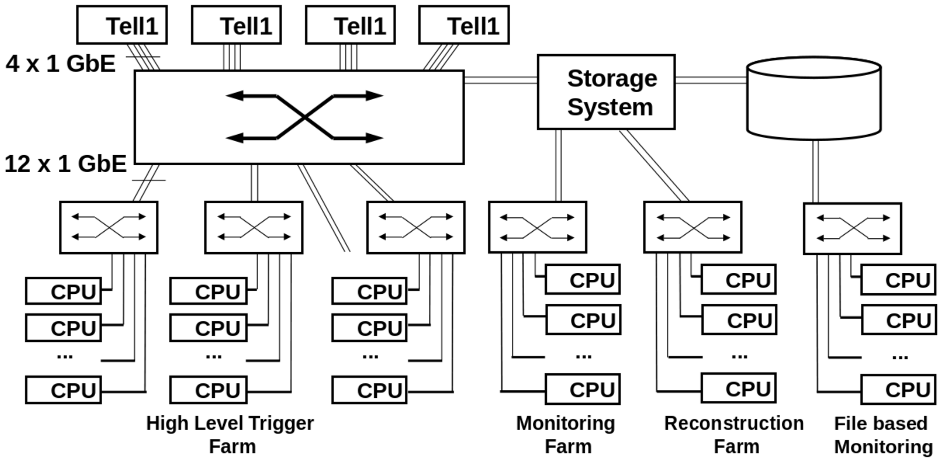

Figure 1. The layout of the LHCb DAQ hardware. Data from particle collisions are sent from the front-end boards (Tell1) through a switching network to the workers of the HLT farm (CPU).

During the 2012 running period the LHC collider delivered particle collisions with stable beams conditions for about 30% of the time, as shown in Figure 2. The remaining time is necessary to ramp the magnets between periods of stable beams, to improve the collider during machine development periods of several days or simply to solve technical problems. During these periods the CPU resources of the HLT farm are not required to operate the LHCb experiment.
In the following sections we present the mechanism to take advantage of the periods without particle collisions and to fully exploit the CPU resources of the HLT farm. The concepts and the configuration of individual nodes is described in section 3, the data monitoring and the controls aspects concerning the operation are discussed in section 4.

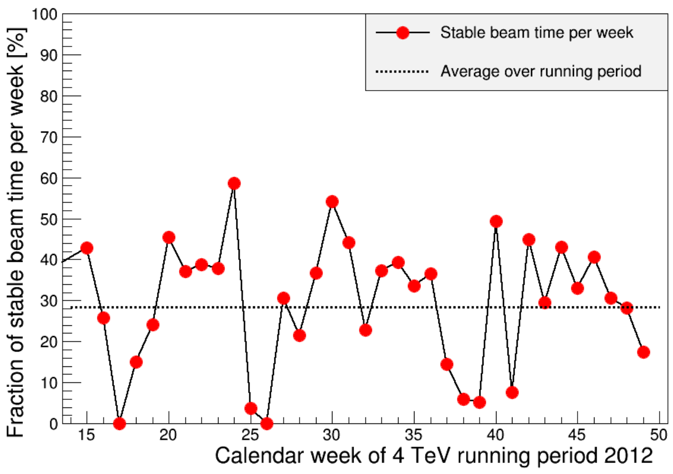

Figure 2. The LHC collider delivers beams useful to collect particle collisions during roughly 30% of the available time (data source: [4]). The dips in the distribution indicate LHC machine development phases typically lasting more than a week. The average was computed over the entire period including machine development periods.

2. Problem Analysis and Solution

Single staged HLT operation is typically synchronous with the delivery of particle collisions: it starts once the beams are declared stable and ends when the beams are dumped. During this period the front-end boards send massive amounts of data to the HLT farm. Here the data must be processed and only a small fraction may be kept for long term storage:

- The front-end boards deliver L0-passed event data with a rate more than 50 GB/s which corresponds to an average input data rate per node of of roughly 30 MB/s.
- The worker nodes are equipped with local disks with a capacity of 2-4 TB, suitable to intermediately buffer the arriving data. The usable size of the intermediate buffer is 6 PB which allows to store data for a period of roughly 30 days assuming a 30% LHC efficiency.
- Splitting the high level trigger program into two stages, HLT1 and HLT2, processing asynchronously allows to decouple the second phase from the delivery of particle collisions.
- The rest are simple budget calculations. Parameters are the CPU consumption per event of HLT1 to be executed in real-time while taking data, the CPU consumption of HLT2 and the size of HLT1 accepted data stored on disk.
- Given the existing resources in LHCb, a HLT1 data reduction rate by a factor of six [5] and a LHC duty cycle of 30% this leads to a execution time of HLT1 of 25 msec and a minimal data buffer time of 30 days. The HLT2 budget would be 330 msec provided there is no loss in switching activities. HLT2 is expected to run constantly on each worker node, but at a lower priority to not negatively interfere with the HLT1 processing.

In the following sections the implementation of the above plan is described with the basic building blocks, their composition and the control thereof by the experiment's control system.

3. Architectural Concepts and Implementation

The independent processing of two event filter stages required a substantial change of the flow of event data in the experiment [6]. The implementation was realized with limited manpower by consequently reusing reoccurring patterns which will be described in the following sections.

3.1. The Buffer Manager Concept

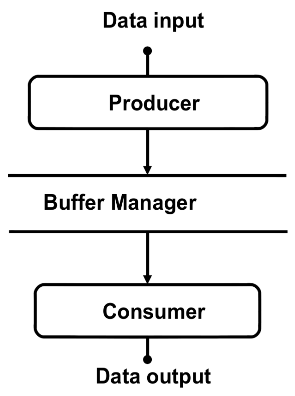

Figure 3. The schematic of the buffer manager concept.

In the LHCb online system all event handling activities are separated into independent, asynchronously executing processes to derandomize the flow of events. This includes all readout functions such as the assembly of events from fragments sent by the TELL1 boards, event filtering or the transport of accepted events to the long term storage. All these functionalities are implemented re-using the basic building block shown in Figure 3 [6]: the producer task is responsible for reading/assembling events from data sources. Once finished, the data block is declared to a managed shared memory area, the Buffer Manager (BM) and the producer is ready for the next read operation. The consumer task is activated each time an event is declared in the BM and is responsible for releasing the space occupied as soon as it finished processing. The two activities of reading and processing can proceed asynchronously, provided there is sufficient space in the buffer to accommodate at least one read operation by the producer. A BM is uniquely identified by its name. Any number of instances may exist on a given worker node.

The BM is managed by a dedicated task executing on each node, which creates and initializes the shared memory area and handles the requests of the producer and consumer applications.

3.2. Data Flow in the Worker Nodes

The buffer manager concept was consistently applied to model the data flow. The left part of Figure 4 shows the dataflow diagram to process the data recorded by the experiment while the LHC beams are in physics settings and saved to the local disk. The data from the TELL1 boards arrive in the worker nodes, get assembled by the event builder *MEPRx* [6]. and placed in the *Events* BM. The data are picked up by the first stage filter processes HLT1 (*Moore1*). The number of filter process instances per node is configurable and depends on the actual hardware. The accepted events are declared to the output buffer *Output* and written to the local disk buffer by the *DiskWriter*. An integrated rate of events of 1 kHz or about 1 Hz per worker node is sent to the monitoring facility to detect possibly malfunctioning detector components.
To not overflow the local disk buffer of the worker node (see Figure 4) the stored data is reduced by the continuously running second stage of the event filter process, HLT2, which execute more sophisticated algorithms and hence requires more CPU resources - though at lower priority. The HLT2 activity is independent from the actual data acquisition process. With respect to the event data flow it is a parallel data taking activity, not with a detector as front-end, but preselected events injected from disk by a data reader process which places the events in the *Events* buffer. The events are consumed by the second level event filter processes HLT2, represented by processes of type *Moore2*. All events accepted are declared to the *Output* BM and sent to the storage system. On each node data are processed in the order they were written i.e. the runs are processed sequentially. The ensemble of all worker nodes however may very well process numerous runs in parallel depending on the data taking history of each worker node.
Figure 4 shows that the HLT1 activity is only connected to the HLT2 activity via the local disk as buffering device. Hence, both activities can be executed and controlled independently

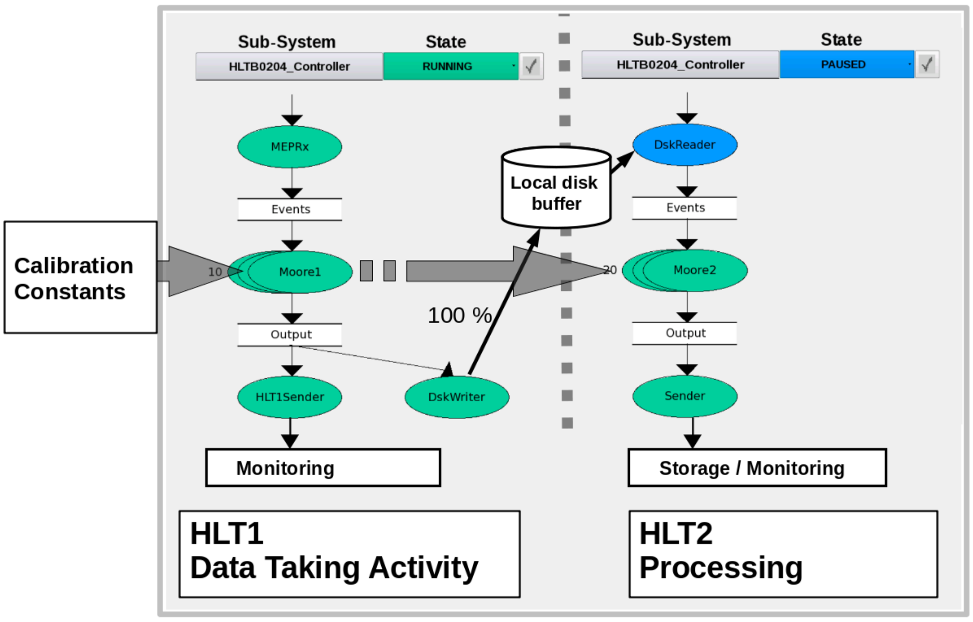

Figure 4. The event data flow in a worker node. The left part of the picture shows the flow of events when taking data from the experiment. The right side of the figure shows the secondary event selection running constantly independent of the LHC delivering particle collisions.

and the data taking process does not impose any fundamental restrictions on the execution of HLT2 unless the the local disk buffer is exhausted. If this happens, the event builders (MEPRx processes in Figure 4) will only be able to allocate space in the *Events* buffer at the rate the HLT2 processing can liberate local disk space. Though the system would not be at halt, this is clearly not a desired working point. To ensure flexibility, the two dataflows for HLT1 and HLT2 may be easily changed using a dedicated editor application. A palette of such configurations allows to quickly reconfigure the processes on each worker node. Both activities, HLT1 and HLT2 use the same calibration constants for the event selection which later are used for offline data processing. The determination of these constants is discussed in the following.

3.3. Online Calibration

The HLT1 and the HLT2 event selection require calibration constants of high quality to select events for offline data analysis and no further offline pass over the accepted events is planned. The calibration constants must be determined from events collected by the experiment at the very moment the events actually should be filtered. As shown in Figure 5 this determines the actions to be performed for data taking during a fill:

- HLT1 processing is purely based on tracking. These constants are computed once all movable devices, such as the vertex detector are in the final position.
- HLT2 in addition uses particle identification. This requires time dependent constants valid for a predefined time period. These are computed on a dedicated computing farm fed from the monitoring data stream of accepted HLT1 events of O(100 Hz). After the data taking activity of each run with a duration of typically one hour finished, the *Particle Identification Calibration* is started. HLT2 processing starts at the earliest after these calibration constants are present.

At the beginning of a fill HLT1 processing is based on tracking constants from the previous fill and a set of events optimized for the *Tracking Alignment* is collected. Once a sufficient amount of suitable events are collected, the alignment process runs in parallel to the data taking activity on the HLT farm. The results obtained on each node are then combined in an appropriate manner and published to all HLT1 processes, which automatically load them following a run-change.

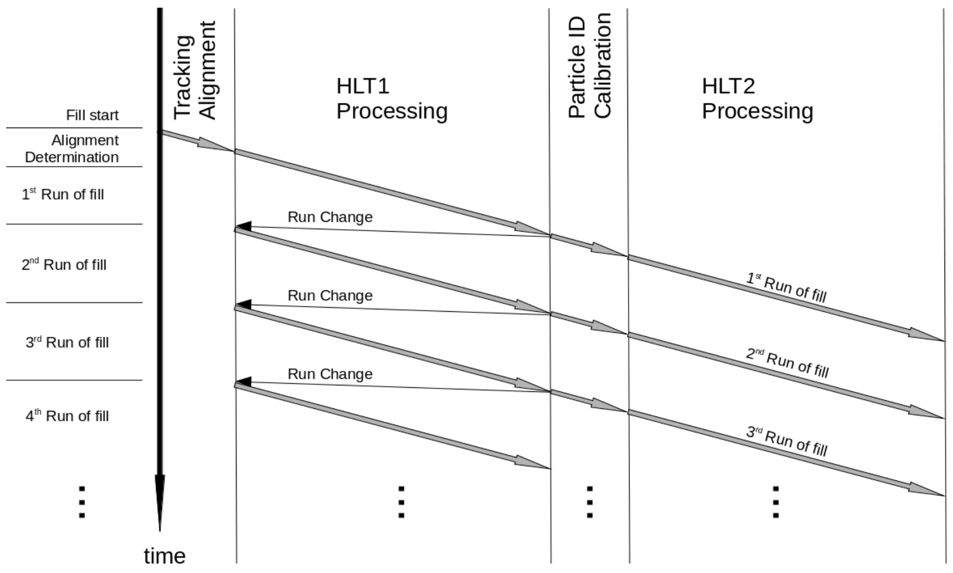

Figure 5. The time ordering of the actions to be performed during a fill. HLT1 processing is synchronous with the LHC collider delivering particle collisions, the start of the HLT2 processing only indicates the earliest starting point: the end of the *Particle ID Calibration*.

4. The Control Hierarchy

The control of all participating equipment and all the processes, which belong to a given activity is performed by a tree-like hierarchy. For a given activity all processes in an HLT node are grouped, like hardware components of the experiment [6].
As elaborated in previous sections, at any time up to three activities may operate in parallel in the HLT farm as shown in Figure 6:

- The *data taking* activity combining the experiment control and the first level event selection. This includes low level monitoring of the detector response in the monitoring farm and higher level monitoring based on reconstructed events in the reconstruction farm. In the reconstruction farm the necessary input data to calibrate the particle identification are collected during each run.
- The *second level event filtering* processes the HLT2 algorithms based on particle collisions written to the local disks. Accepted events are transferred to long term storage for later physics analysis. The HLT2 control infrastructure ensures that no HLT2 event processing is started in the absence of proper calibration constants.
- The *calibration and alignment activity* determining the constants for tracking or for particle identification.

The control of the simultaneously executing activities was realized with three independent tree hierarchies – one for each activity. From the controls aspect these trees are nearly identical though of course only one controls the experiment hardware.

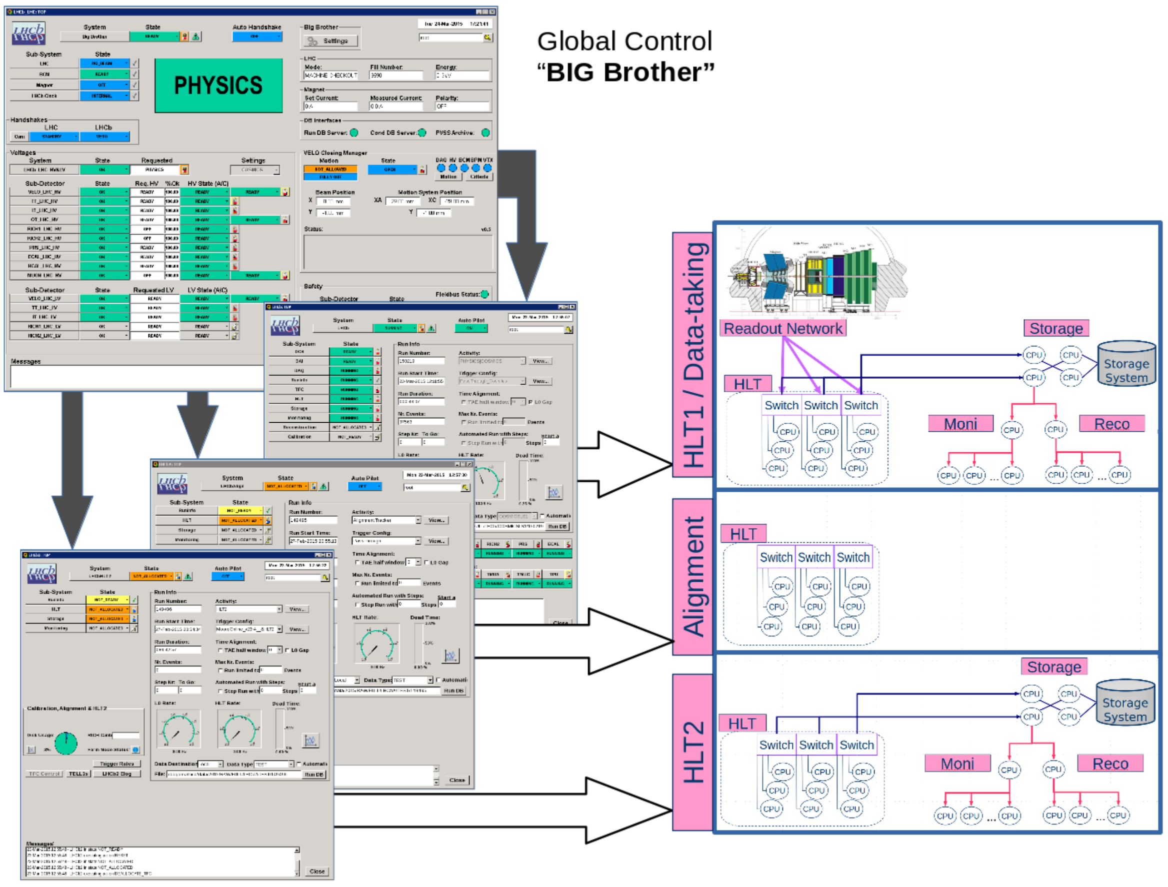

Figure 6. The overall experiment control with three distinct controls-trees: To the left the experiment control Big Brother in charge of driving the interfaces controlling the three activities described above.

At the highest level the operator issues commands to each activity responsible for the overall orchestration of the state transitions. All control hierarchies expose to the operator only macroscopic entities like the experiments data acquisition hardware, the HLT farm and optionally a partition for data monitoring, data reconstruction or the data storage system.
Obtaining the calibration constraints complicates data taking. To ease life and to minimize the possibilities of mis-operation by the human operators these actions may also be issued in automatic manner by the experiment control system. This top level FSM application called *Big Brother* controls the sequencing of the three activities in the appropriate order.
The controls hierarchy was implemented using the commercial SCADA system WinCC-OA [9], and SMI++ [10], a state manager interfaced to WinCC-OA. All event data processes in the LHCb online system are based on the LHCb data processing framework Gaudi [11]. They accept transition requests from SMI++ and publish their state to SMI++ using the DIM protocol [12].

5. Data Monitoring

The HLT1 and HLT2 activity differ in nature: HLT1 processes at any time exactly one run, whereas HLT2 may handle several runs due to the asynchronous processing in the worker nodes. This leads to activity specific solutions for data monitoring:

- Low-level monitoring to ensure the correct functioning of the hardware components of the experiment and the operations aspects of the HLT1 selection process is based on events accepted by HLT1 . Monitoring is performed in the *Monitoring farm* [7]. A fraction of these events is reconstructed at run-time in the *Reconstruction farm* for high-level monitoring using tracks and vertices as shown in Figure 1.
- Monitoring the correct behaviour of the HLT2 algorithms on a per run basis can only be based on files containing HLT2 accepted events. The activity is performed in the file based monitoring facility. HLT2 monitoring application only uses a limited number of events and starts as soon as some worker nodes have started HLT2 processing. The implementation uses the buffer manager concept as described in section 3.1. Event producers, which in this case may either be file readers of network receivers declare events to the buffer managers. Event consumers, here either the event senders distributing the data to the worker nodes or the monitoring workers declare interest to the buffer manager and consume the events.
- Though not activity related, at the end of the HLT2 processing the physics performance is monitored. This processing pass, which during *Run1* was a purely offline execise is now done online, using a representative fully reconstructed sample of events corresponding to O(100 Hz) throughout each run. In a dedicated facility as shown in Figure 1 roughly 60 instances of

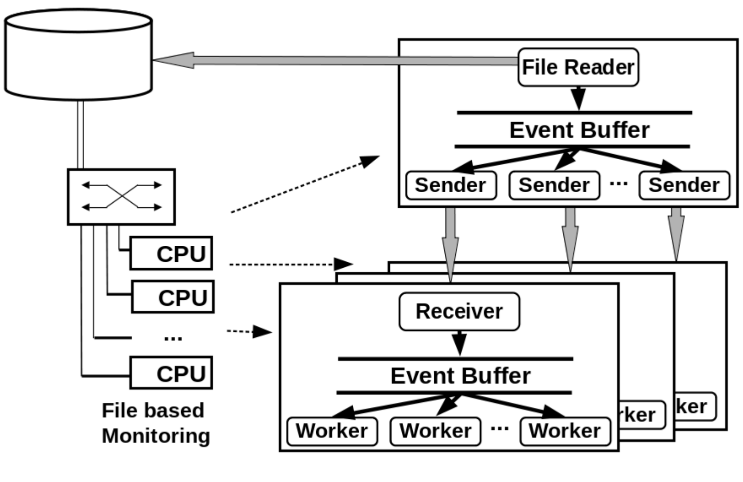

Figure 7. The implementation of the file based monitoring cluster.

the monitoring processes are fed by a reader process while data files are still accessible on the disk storage at the experiment site. The processes verify the proper software configuration by checking distributions using selected physics channels. The data distribution by the reader to the workers use the Buffer Manager pattern described in section 3.1. The control of this entity is implemented similar to a subfarm in the HLT using WinCC-OA.

6. Conclusions

The buffer manager technique and a flexible controls system design supporting multiple parallel data acquisition activities allowed to implement a very flexible event filter architecture. This architecture together with local event buffering on the worker nodes will help to extend the usable CPU time of the HLT farm for physics event filtering beyond the time the experiment actually will receive particle collisions from the LHC collider. The expected gain of usable CPU time of a factor 3 depending on the collider performance will help in the 2015 running period to improve the physics results. The gain will be achieved by separating the event filtering into a fast and a slower component executing asynchronously. Special effort was put on extensive data monitoring facilities at various stages of the event filtering process to ensure the proper functioning of the detector and the software.

References

[1] LHCb Collaboration, "LHCb the Large Hadron Collider beauty experiment, reoptimised detector design and performance", CERN/LHCC 2003-030
[2] A. Tsaregorodtsev et al., "DIRAC: a community grid solution", 2008 J. Phys.: Conf. Ser. 119 062048.
[3] E. van Herwijnen, "Control and Monitoring of on-line trigger algorithms using Gaucho", International Conference on Computing in High Energy and Nuclear Physics (CHEP 2006), Mumbai, India, 2006; proceedings.
[4] LHC fill summary table, stable beam duration for 4 TeV running 2013, http://lhc-statistics.web.cern.ch/LHC-Statistics/index.php.
[5] J. Albrecht, G. Raven, V. Gligorov, private communication.
[6] M. Frank et al., "The LHCb High Level Trigger Infrastructure", International Conference on Computing in High Energy and Nuclear Physics (CHEP 2007), Victoria, BC, Canada, proceedings.
[7] O. Callot et al., "Data Monitoring in the LHCb Experiment", International Conference on Computing in High Energy and Nuclear Physics (CHEP 2007), Victoria, BC; proceedings.
[8] C. Gaspar et al., "Partitioning in LHCb", LHCb 2001-116 DAQ.
[9] SIMATIC WinCC Open Architecture [Online], http://www.pvss.com
[10] C. Gaspar et al., "SMI++ - Object Oriented Framework for Designing Control Systems for HEP Experiments", International Conference on Computing in High Energy and Nuclear Physics (CHEP 1997), Berlin, Germany, 1997; proceedings.
[11] G. Barrand et al, "GAUDI : The software architecture and framework for building LHCb data processing applications", Comput. Phys. Commun. 140 (2001) 45-55.
[12] C. Gaspar et al., "DIM, a Portable, Light Weight Package for Information Publishing, Data Transfer and Inter-process Communication", International Conference on Computing in High Energy and Nuclear Physics (CHEP 2000), Padova, Italy; proceedings.