The LHCb Online system in 2020: trigger-free read-out with (almost exclusively) off-the-shelf hardware

T Colombo[1], A Amihalachioaei[1,2], K Arnaud[3], F Alessio[1], L Brarda[1], J-P Cachemiche[3], D Cámpora[1,4], S Cap[5], L Cardoso[1], F Cindolo[6], M Daoudi[1], P Durante[1], P-Y Duval[3], C Faerber[1], P Fernández[1], M Frank[1], D Galli[6], C Gaspar[1], F Hachon[3], M Jevaud[3], B Jost[1], R Le Gac[3], M Manzali[7], U Marconi[6], H Mohammed[1], N Neufeld[1], F Pisani[1,6], L Promberger[1,8], C Quast[1], F Réthoré[3], F Sborzacchi[9], R Schwemmer[1], S T'jampens[5], S Valat[1], J Viana Barbosa[1], B Vőneki[1] and G Vouters[5]

[1] CERN, Geneva, Switzerland
[2] Gheorghe Asachi Technical University, Iași, Romania
[3] Centre de Physique des Particules de Marseille (CPPM), Marseille, France
[4] Universidad de Sevilla, Sevilla, Spain
[5] Laboratoire d'Annecy-le-Vieux de Physique des Particules (LAPP), Annecy-le-Vieux, France
[6] Università di Bologna and Sezione INFN, Bologna, Italy
[7] Università di Ferrara and Sezione INFN, Ferrara, Italy
[8] Hochschule Karlsruhe – Technik und Wirtschaft, Karlsruhe, Germany
[9] Laboratori Nazionali di Frascati dell'INFN, Frascati, Italy

E-mail: `Tommaso.Colombo@cern.ch`

Abstract. The LHCb experiment at CERN has decided to optimise its physics reach by removing the first level hardware trigger for 2020 and beyond. In addition to requiring fully redesigned front-end electronics this design creates interesting challenges for the data-acquisition and the rest of the online computing system. Such a system can only be realized within realistic cost using as much off-the-shelf hardware as possible. Relevant technologies evolve very quickly and thus the system design is architecture-centred and tries to avoid to depend too much on specific technologies. In this paper we describe the design, the motivations for various choices and the current favoured options for the implementation, and the status of the R&D. We will cover the back-end readout, which contains the only custom-made component, the event-building, the event-filter infrastructure, and storage.

1. Introduction

LHCb [1] is one of the four major experiments at CERN's Large Hadron Collider (LHC). It is a single-arm forward spectrometer, designed for precision measurements of B-meson decays. In 2019 and 2020 the LHCb experiment will be substantially upgraded in order to reach unprecedented precision on the main observables of the *b* and *c*-quark sectors. Critical parts of this upgrade are the trigger-less readout system and the full software trigger.

The experiment observes proton-proton inelastic collision events delivered by the LHC at a rate of 30 MHz. Due to limitations in the current readout system, this rate must be reduced to 1.1 MHz using a fixed-latency hardware trigger system [2], which can only base its selection on basic physics signatures available at this stage (high-energy or high-momentum particles). The selection is further refined by a software event filter (also called high-level trigger).

The hardware trigger is currently the largest source of event selection inefficiencies for the experiment. Therefore, one of the main goals of the LHCb upgrade is its replacement with a full software event filter. This requires a new trigger-less readout system, a new event builder, and a substantially upgraded event filter system in order to acquire, assemble, and select events at the full rate of 30 MHz. The upgrade enables applying event selections that are as similar as possible to those used in offline analyses, so that selection efficiencies are maximised and systematic uncertainties are minimised [3].

The building blocks of the new system are shown in Figure 1. The front-end electronics of the various detectors are synchronised with the LHC collision frequency by the timing and fast control system (TFC). After zero-suppression, the acquired data fragments are pushed to around 500 event builder servers via almost 10000 point-to-point optical links. Event builder nodes are all interconnected by a dedicated high-performance network used to assemble event fragments into full events. They are divided in small groups, with each group sending the events it assembles to a subdivision of the event filter computer farm (called sub-farm). The event filter nodes run the selection algorithms. They have access to a large disk-based temporary storage for events, that is used to relax the throughput requirements on the event filter and to maximise the farm utilisation.

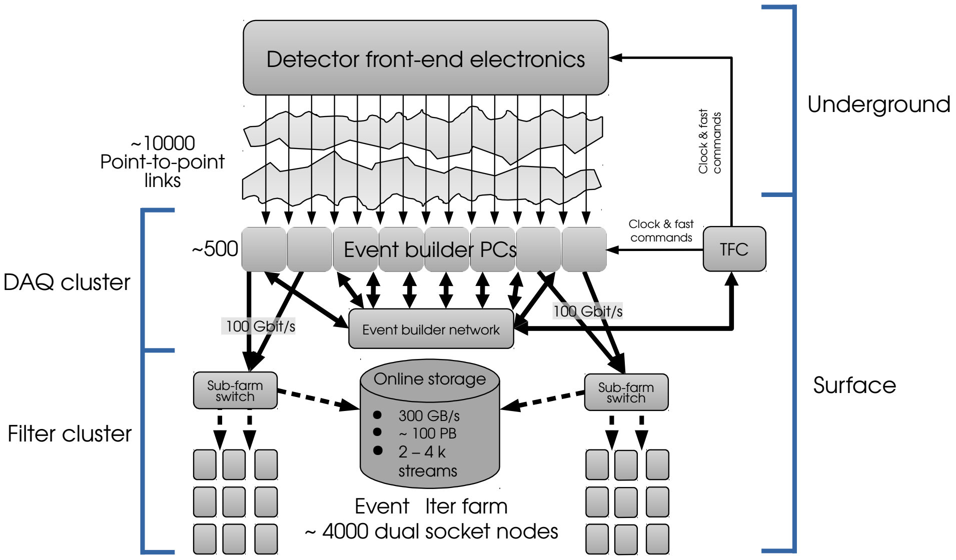

Figure 1. Layout of the new LHCb DAQ & event filter.

2. Readout, timing, and fast control

Each detector in the upgraded experiment will be equipped with front-end electronics capable of acquiring its signals at the LHC maximum collision rate of 40 MHz [1]. The interface between the detectors and the readout system consists of almost 10000 simplex optical links. The links use a custom protocol (GBT [4]) developed to be implementable with radiation-hardened electronics, a necessary feature to operate in close proximity with the detectors. Each optical link can carry up to 4.48 Gb/s of data from the underground experimental area to the surface where it is received by a custom PCI Express readout board, called TELL40, housed in an event-builder node. The received event fragments are buffered in the node's main memory waiting to be assembled into a full event. A dedicated supervisor board, called SODIN, centrally manages the readout of events. It distributes the LHC clock to the rest of the system, it ensures that all front-ends and readout boards are synchronous, and it regulates the data-flow from the front-ends to the readout boards. Dedicated interface boards, called SOL40, fan-out the timing and synchronisation commands from the supervisor board to the front-ends and readout boards, while also acting as the interface between the front-ends and the experiment's control system. The flow of event data and commands is shown in Figure 2. More details on the clock and timing distribution are available in [5].

The functionality of the readout boards (TELL40), supervisor board (SODIN), and interface boards (SOL40) is actually implemented on the same hardware platform: a PCI Express Gen 3.0 x16 add-in card, shown in Figure 3, based on an Intel Arria 10 FPGA and high-density optical

[1] In LHCb, one every four "collision" events delivered by the LHC is actually empty, so the real inelastic collision rate corresponds to the 30 MHz quoted in section §1.

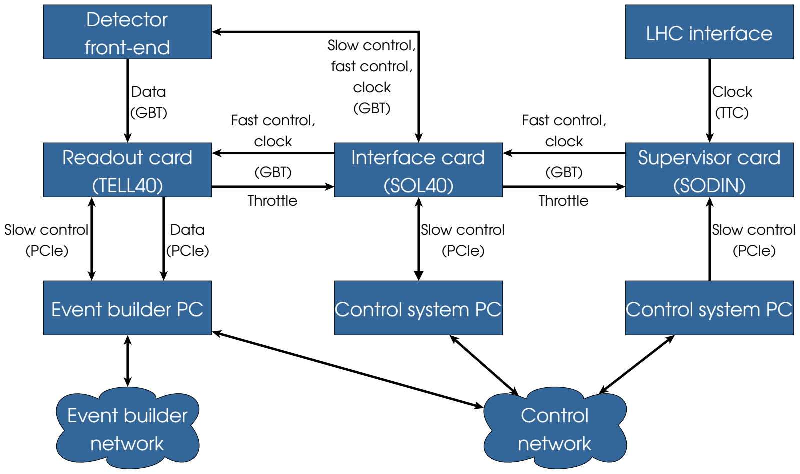

Figure 2. Readout, timing, and fast control data-flow.

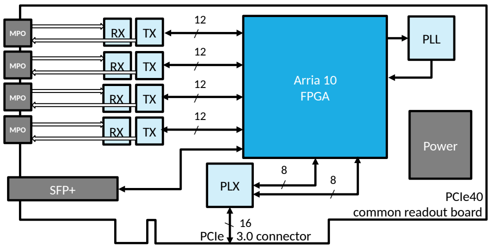

Figure 3. The common LHCb readout, timing, and control board.

I/O with up to 48 duplex GBT ports. The functional differentiation is achieved by changing the card's firmware. These cards are the only pieces of custom-built hardware in the system. In order to read out the entire LHCb experiment, 500 readout boards, less than a hundred interface boards, and a few supervisor boards are necessary. Having all three functions share a single hardware platform lowers their production and maintenance costs.

In its readout incarnation, the FPGA is programmed with a high-performance DMA engine capable of copying the received event fragments into the main system memory at over 100 Gb/s. More details on its architecture and performance are available in [6, 7].

3. Event builder

The main goal of the upgraded LHCb data-acquisition system is enabling the event filter to efficiently select events based on the full data readout from the detectors. This requires designing an event builder system capable of collecting and aggregating data fragments from the various detectors at the full collision rate of 30 MHz. The data size of a full event is projected to be as high as 150 kB. Therefore, taking into account protocol overheads and other potential inefficiencies, the design goal for the new event builder was set to a sustained throughput of up to 40 Tb/s.

An event builder is essentially made up of three components: the readout units, the builder units, and the network interconnecting them. Each readout unit receives the event fragments from the detector point-to-point links and makes them available on the network. For each event, one of the builder units gathers all the event fragments from all the readout units, assembles them into a complete event, and makes it available to the event filter. Since the data flows in a single direction (from the readout units to the builder units), it is possible to double the network utilisation and halve the total number of nodes in the event builder system by "folding" the builder units onto the readout units. Essentially, a single event builder node can host both a readout unit, which outputs event fragments onto the network, and a builder unit, which receives event fragments from the same network. An obvious problem with this architecture is that, if the event builder network is already close to saturation, a separate network is necessary to transfer the complete events from the builder units to the event filter nodes. However, this is not necessarily a disadvantage: being single-purpose, the event-builder network can be based on any high-performance network technology: the choice depends solely on the price/performance ratio for this specific application. Moreover, the traffic pattern is known in advance and it can be tuned to maximise network usage. For these reasons, the development effort for the new LHCb event builder is focused on the folded architecture.

In principle, the basic readout/builder unit can be implemented by a standalone FPGA-based readout board equipped with the appropriate number of inputs, to receive data from the experiment and from the event-builder network, and outputs, to send data to the event-builder network. Ostensibly, dispensing with the requirement for a server to host the readout boards would lead to significant cost savings. As a matter of fact, this solution comes with significant limitations. High-performance network technologies, such as InfiniBand and OmniPath, or a full TCP/IP over Ethernet stack are generally too complex or expensive to implement on a FPGA. While there are numerous commercially available integrated circuits implementing these technologies, these are commonly designed to interface with a complete computer and operating system via PCI Express links. Moreover, purely FPGA-based network stacks must be relatively limited, to save on FPGA resources and complexity, and most likely cannot include a network congestion control mechanism. This must be compensated by adequately large buffers in the network switches interconnecting the boards. Deep-buffer network switches are a niche product and as such tend to be very expensive compared to their shallow-buffered counterparts. These and other considerations, explained in [8], led to the adoption of an event builder architecture based on commercially available servers and network interfaces.

The basic building block for the implementation is an event-builder node housing three PCI Express x16 add-in cards: a TELL40 readout board, a 100 Gb/s network adapter for the event builder network, and another 100 Gb/s network adapter to send the assembled events to the event filter nodes for processing. The total number of nodes is essentially determined by the number of TELL40 readout boards needed by the various detectors, currently estimated to be around 500. In order to reach the required 40 Tb/s aggregate throughput, each node must be able to sustain a throughput of at least 80 Gb/s. Two measurements are crucial to achieving that goal: internal node throughput and aggregate network throughput.

Each node must be capable of handling a 80 Gb/s flow from the TELL40 to its main memory and from there to the event-builder network adapter, plus another 80 Gb/s flow from the event-builder network adapter to the main memory and from there to the event filter network adapter. This is certainly possible on current-generation server hardware, as shown in [9]. However, at those rates, network protocols implemented in software, as is usually the case for the TCP/IP suite for example, introduce a very significant CPU usage both for packet processing and kernel-space/user-space data copies [9]. Both these problems can be avoided using hardware-assisted remote DMA (RDMA) technologies. The InfiniBand and OmniPath network stacks support this by design; many Ethernet network adapters also support either iWARP or RoCE, two competing approaches to RDMA over Ethernet.

Having established that a single node is capable of sustaining the required throughput is not enough: the event-builder network must be able to sustain the aggregate traffic to and from the nodes. The traffic pattern is particularly challenging. To maximise throughput, every node builds a different event in parallel with all the other nodes. This results in a continuous network-wide scatter/gather operation: the readout unit on each node scatters the fragments corresponding to each event to their respective builder units; the builder unit on each node gathers all the fragment corresponding to a single event from the readout units that have them. Naturally, a bottleneck-free network would be best suited to handle this all-to-all traffic. Ideally, if the nodes are partitioned in two arbitrary halves, the bandwidth available between the partitions should always correspond to the sum of the bandwidths of one half of the nodes. The most commonly network topology with this property is the so-called fat tree: a multi-layer network with equal link capacity between the layers, implemented by building a tree with multiple roots. The simplest incarnation of a fat-tree network, consisting of two layers of 4-port switches, is shown in Figure 4.

Even in the absence of bottlenecks, maximising the network usage is not an easy task: collisions, i.e. two or more senders trying to use the same network link at the same time, cause inevitable performance degradation. An extensive measurement campaign is currently underway to demonstrate the feasibility of a 40 Tb/s event builder with a fat-tree network. Fortunately, many existing supercomputers employ a fat-tree network, so a proof-of-concept requires only the development of event builder benchmark software, but no hardware investment. Two such benchmarks are currently in development: DAQPIPE [9] and LSEB [10]. They can probe a large parameter space, including varying communication sizes, number of in-flight communications,

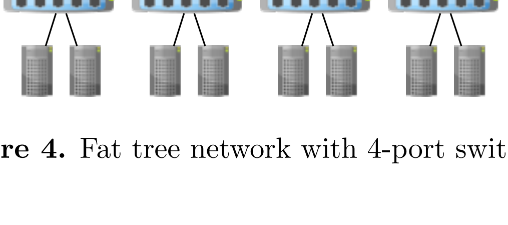

Figure 4. Fat tree network with 4-port switches.

and different communication scheduling patterns. Among those, the most relevant

The results of the largest-scale test performed so far, on a 72-node partition of an 100 Gb/s InfiniBand-based supercomputer with 18 nodes per switch, are shown in Figure 5. Two different communication scheduling patterns were used: a “shift” pattern (each node H_i sends a block of data to node H_{i+1}, then the next block to H_{i+2}, and so on until all other nodes have received data from H_i), and a “random” pattern (where each node sends a block of data to a randomly chosen destination until all other nodes have received data from it). With the random pattern, and an optimised selection of its other parameters, the DAQPIPE benchmark can achieve a per-node bandwidth of around 72 Gb/s, 8 Gb/s short of the stated objective of 80 Gb/s per node.

While the benchmark results are promisingly close to the goal, they also cast a doubt on the scalability of the system: Figure 5 shows that, as the number of nodes in the system increases, the per-node throughput decreases. This is most likely due to the increased probability of communication collisions as the network grows. Performing a continuous scatter/gather collective operation on a fat-tree network while completely avoiding collisions is possible [11], but requires all nodes to schedule their communications in a synchronised fashion. A proof-of-concept test of this idea was performed on 32 nodes interconnected by a 1 Gb/s Ethernet fat-tree network with 4 nodes per switch. With the nodes’ clocks synchronised using the NTP protocol and a scheduling granularity of 100 ms, large enough to absorb the effects of operating system jitter, each node can reach 90% of the maximum throughput of its network interface. These results suggest that this distributed scheduling is a viable way of increasing the event builder’s performance. However, the tests should be repeated at a larger scale with high-performance network interfaces to fully validate the concept.

4. Event filter

The event filter computing cluster will consist of as many as 4000 dual-processor servers. It will employ a two-level filtering strategy already in use and validated with the current experiment. A first filter level, called HLT1, will perform a fast reconstruction and selection of the events at the full data-acquisition rate of 30 MHz. It is expected to reduce the event rate to around 1 MHz. The events selected by HLT1 will be saved to a temporary disk buffer. A second filter level, called HLT2, will then asynchronously read the saved events and perform a full reconstruction and selection, reducing the event rate to a final output of around 100 kHz.

The intermediate disk buffer allows the maximal exploitation of the event filter cluster. The LHC isn’t constantly providing collisions: after a collision period that can last up to a day, a few hours are needed to prepare for the next collision period. Therefore, when the accelerator is

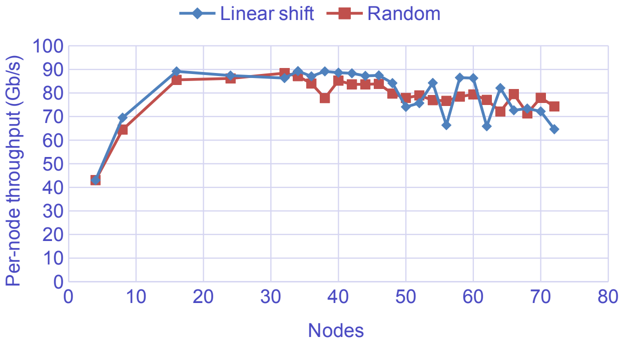

Figure 5. Per-node throughput obtained with the DAQPIPE event builder benchmark on up to 72 nodes, using two different communication scheduling strategies.

running, most of the servers will be dedicated to HLT1 to ensure a timely handling of the new experimental data. During the LHC downtime, all the nodes will instead run HLT2 to finish processing the fraction of events that couldn't be handled in real-time. This strategy keeps the event filter cluster occupied at all times, but requires a large, high-performance storage system: it must sustain a maximum throughput of 150 GB/s in input and 150 GB/s in output. Moreover, to absorb long LHC collision periods, its capacity should be in the order of tens of petabytes. Its definitive implementation is not yet defined and will be driven by cost considerations.

5. Conclusion

The LHCb experiment will undergo an ambitious upgrade in 2019 and 2020. Its readout system will be upgraded to support trigger-less data acquisition at 30 MHz, which will require an event builder system capable of a throughput of 40 Tb/s.

The execution of the upgrade plans is proceeding well. The readout boards, their firmware, and associated control software are already well advanced in development. The event builder benchmarks present no show-stoppers and tests suggest that there is room for further improvement. Implementation evaluations are underway for the event filter cluster and its associated network and storage.

Big implementation challenges remain. The upgrade project is entering its most interesting phases yet: the finalisation of designs and finally the construction of the system.

References

[1] LHCb Collaboration 2008 *JINST* **3** S08005

[2] LHCb Collaboration 2003 *LHCb trigger system Technical Design Report* CERN-LHCC-2003-031 (Geneva: CERN)

[3] LHCb Collaboration 2014 *LHCb Trigger and Online Upgrade Technical Design Re*port CERN-LHCC-2014-016 (Geneva: CERN)

[4] Moreira P, Marchioro A and Kloukinas K 2007 *Proc. Top. Workshop Electron. Part. Phys.* (Geneva: CERN) p 332

[5] Alessio F *et al* 2015 *JINST* **10** C02033

[6] Bellato M *et al* 2014 *J. Phys.: Conf. Ser.* **513** 012023

[7] Durante P *et al* 2015 *IEEE Trans. Nucl. Sci.* **62** 1752

[8] Schwemmer R and Neufeld N 2015 *IEEE Trans. Nucl. Sci.* **62** 1747

[9] Valat S *et al* 2017 *IEEE Trans. Nucl. Sci.* **64** 1480

[10] Manzali M *et al* 2017 *IEEE Trans. Nucl. Sci.* **64** 1486

[11] Zahavi E *et al* 2011 *Concurr. Computat.: Pract. Exp.* **22** 217