Large-Scale DAQ Tests for the LHCb Upgrade

Matteo Manzali, Antonio Falabella, Francesco Giacomini, Umberto Marconi,
Niko Neufeld, Sèbastien Valat, and Balazs Voneki

***Abstract*— To increase the event yield, the LHCb experiment will undergo a major detector upgrade planned during the second long shutdown of the Large Hadron Collider (2019–2020). The new data acquisition has to process the whole 40-MHz input event rate, relying only on the large-scale computing farm implementation of the high-level trigger. Event fragments will be forwarded at 40 MHz from the detector front-end electronics to the event builder (EB), through optical links and peripheral component interconnect express cards. The EB farm, of about 500 computers, shall provide an aggregated throughput of 32 Tb/s. To reach the required EB performance, we are testing various interconnect technologies and network protocols on large-scale computing clusters. For this purpose, we have developed an EB software evaluator. We report here about the results of the measurements performed on high-performance computing facilities to test throughput and scalability.**

***Index Terms*— Data acquisition (DAQ), event building, high-throughput computing, infiniBand, LHCb.**

I. Introduction

The LHCb experiment is one of the four main experiments running at the Large Hadron Collider (LHC) at CERN. It aims at performing very precise measurements of charge conjugation parity symmetry violation and rare decays of *b* and *c* quark hadrons, collecting unprecedented data samples. For this purpose, the upgraded triggerless [1] readout system will operate at the bunch crossing frequency of 40 MHz. The event builder (EB) is the key component of the new readout system. It collects and reassembles the event fragments delivered by the subdetector readout boards, henceforth called PCIe40 [2]. Each PCIe40 is equipped with up to 24 optical links coming from the detector and it is directly connected to a node of the EB through a peripheral component interconnect express (PCIe) slot. Consecutive event fragments transmitted from the front-end electronics are received and buffered by the PCIe40 and then copied into the EB node memory by means of direct memory access (DMA) operations. The PCIe readout requires therefore about 500 PCIe40 cards to read out the whole detector and the same number of EB nodes, assuming just one PCIe40 card connected to one node [3]. In order to acquire the desired bandwidth of 32 Tb/s with an EB network composed of 500 nodes, each PCIe40 card will write at an expected bandwidth of 100 Gb/s. The EB will be implemented using a high-throughput low-latency interconnect, using off-the-shelf hardware. Studies are ongoing to establish which is the appropriate network technology to use [4]. The present candidates available on the market are InfiniBand [5], 100 G Ethernet [6], and Intel Omni-Path [7]. The results presented here refer to InfiniBand, presently the most mature technology, as demonstrated by its wide adoption in the high-performance computing field. This paper is organized as follows. The EB software we have developed is described in Section II. Section III describes some system tuning options to optimize the performance. Results of the large-scale tests are shown in Section IV. In Section V, we report about the results of some experimentation with low-power boards based on the x86 architecture. Finally, in Section VI, we summarize our conclusions.

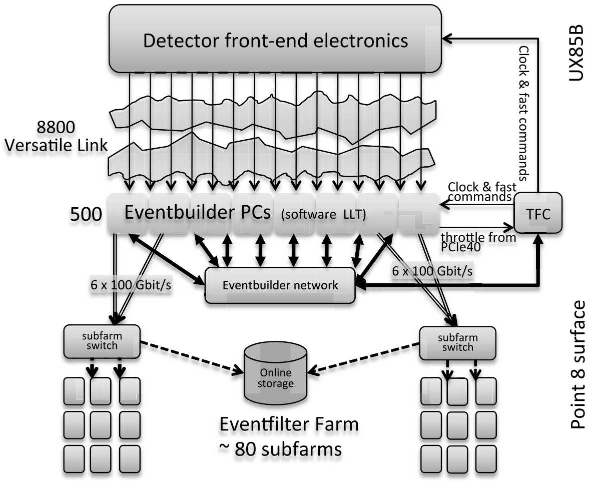

Fig. 1. Architecture of the upgraded LHCb readout system.

II. Event Builder Software

The schema of the upgraded readout system is shown in Fig. 1. The central part of the readout system is the EB cluster that will consist of about 500 nodes interconnected with a high-throughput network technology able to sustain data transfers at a rate of 100 Gb/s in both directions (input and output). Each EB node includes two distinct logical components: the readout unit (RU) and the builder unit (BU). An RU receives event fragments from the detector and ships them to a receiving BU in a many-to-one pattern. Each BU gathers the event fragments and assembles them in full events, which are then sent out to event filter farms (EFFs) for processing.

A. Main Components

The EB software implementation described in this paper is called *large-scale EB* (LSEB) [8]. In order to keep the

Manuscript received June 29, 2016; revised January 26, 2017, March 29, 2017, and April 3, 2017; accepted April 8, 2017. Date of publication April 12, 2017; date of current version June 26, 2017.

M. Manzali is with the University of Ferrara, 44121 Ferrara, Italy (e-mail: matteo.manzali@cnaf.infn.it).

A. Falabella and F. Giacomini are with INFN-CNAF, 40127 Bologna, Italy.

U. Marconi is with INFN-Bologna, 40127 Bologna, Italy.

N. Neufeld, S. Valat, and B. Voneki are with CERN, 1211 Geneva, Switzerland.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNS.2017.2693422

communication management separated from the logic of the event-building, LSEB can be nominally split into two distinct layers, namely, the communication layer and the logic layer. The communication layer includes primitives for data communication between nodes and relies on the InfiniBand *verbs* interface [9], a library that offers a user-space application programming interface to access the remote DMA (RDMA) capabilities of the network device. On top of the communication layer sits the logic layer, a set of software components performing the actual event-building under realistic conditions. The main components of the logic layer are the following:

1) Controller: The *Controller* is the software component that simulates the acquisition of data coming from the detector. The Controller handles two distinct memory ring buffers, one for the event fragments and one for some accompanying metadata that is foreseen in order to speed up the management of the data. The buffers are prefilled with randomly generated data when the program starts in order to avoid possible delays at runtime caused by the data generation. However, the Controller implements the full protocol to reserve and release the data. The real fragments coming from the detector will not in general have a fixed size, being collision events different from each other. The Controller simulates this behavior assigning a random size to each fragment.

2) Readout Unit: The aim of the RU is to send ready fragments to the selected BU at the maximum speed allowed by the network. The minimum messages size needed to saturate the available bandwidth mostly depends on the network device. In fact, different nominal data rates or even equal nominal data rates but with host channel adapters (HCAs) from different vendors may require different message sizes to saturate the bandwidth. Since the size of a single fragment, which is of the order of a few hundreds of bytes, may not be enough to saturate the available bandwidth, the RU sends fragments in bulks of a configurable number of fragments. The RU is aware of the memory areas containing metadata and fragments. However, the information exchange between the RU and the Controller is limited to the metadata, avoiding the overhead caused by the access to the whole fragments in memory. This approach allows also a rapid identification of the beginning and the end of the memory area containing the fragments to be sent to a BU, without reading each fragment.

3) Builder Unit: The aim of the BU is to receive fragments by all the RUs and to forward them as complete events to the event filter farm. In LSEB, this last step is not implemented yet. A potential effect of the forwarding of complete events to the EFF could be an increase of the memory bus contention, but this aspect and its possible implications are currently under study. Instead of sending the complete events to the EFF, the BU simply checks the correctness of each fragment and then releases the corresponding data buffers.

B. Scheduling Strategy

In the adopted EB logic, the RUs know which BU has been assigned to a specific event, so they send out messages without waiting for a request from the BUs; this is the so-called *push* approach. The design could foresee a supervisor, which would decide how to assign events to the BUs. A supervisor would need to maintain an overall state of the system and keep it up-to-date. It would also need to continuously contact the RUs to communicate which BU has been elected for each event or set of events. Its presence would then cause overhead in terms of software complexity, latency, and network traffic. We decided to follow, at least until the need for a central load-balancing mechanism is demonstrated, an alternative approach, whereby a predefined scheduling strategy is adopted by all the RUs. The strategy is a simple round-robin pattern, in which the ith fragment is sent to the jth BU with $j = i\%N$, where N is the number of nodes of the EB. This choice has the additional advantage of avoiding any single point of failure. The disadvantage of this supervisorless approach is that if a node fails, besides not receiving fragments from the corresponding RU, the corresponding BU will not build its share of events, which will then be lost.

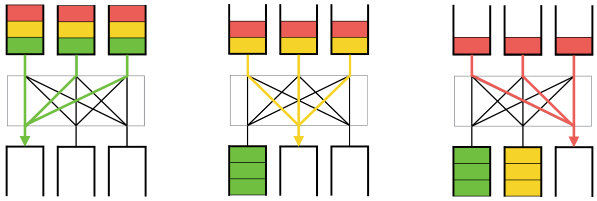

Fig. 2. Schematic view of the push protocol: the RUs are concurrently sending fragments to the same BU.

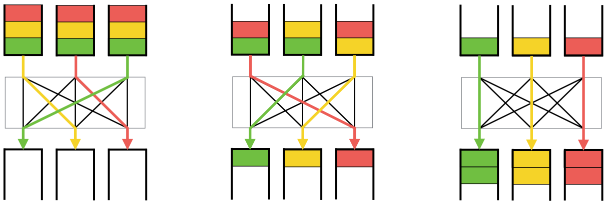

Fig. 3. Schematic view of the push protocol with simple traffic-shaping: the RUs are concurrently sending fragments to different BUs.

C. Dispatching Policy

The push approach adopted by LSEB allows a linear and simple dataflow, but there is the risk that all the RUs send fragments to the same BU almost at the same time, as shown in Fig. 2. This may produce a congestion somewhere in the network infrastructure or, more likely, the software on the receiving side may not be fast enough to serve all the incoming requests. One possible mitigation to these limitations is to introduce a traffic-shaping strategy, aiming at the full network occupation. Therefore, LSEB implements a custom dispatching policy, shown in Fig. 3: all nodes having a unique identifier, each RU starts to send fragments to the BU with an identifier immediately following its own and proceeding in a round-robin order. The local BU will be the last one to receive fragments. The drawback of this approach is that, even if an RU has fragments ready to be sent, it may have to wait, leading to a higher memory consumption. However, considering that the average size of an event is about 100 kB and that the PCIe40 readout board can handle up to 4 GB of the host memory, the size of the fragment buffers is still acceptable.

III. System Tuning

Reaching the maximum attainable performance often requires tuning several hardware features, such as the CPU frequency (CPUFreq), the power management mechanisms, and the process affinity.

A. CPU Frequency Governor

The CPUFreq governor [10] allows the clock speed of the processor to be adjusted on the fly, so that the system can run at a reduced clock speed to save power. The rules for shifting frequencies are defined by the CPUfreq *governor*. The governor defines the power characteristics of the system CPU, which in turn affects CPU performance. Each governor has its own unique behavior, purpose, and suitability in terms of workload.

1) Ondemand Governor: The default is the *ondemand* governor, a dynamic governor that allows the CPU to achieve maximum clock frequency when the system load is high, and the minimum clock frequency when the system is idle. While this allows the system to adjust power consumption according to the system load, it does so at the expense of latency due to the frequency switching. As such, latency can offset any performance and power saving benefits offered by the ondemand governor if the system switches between idle and heavy workloads too often.

2) Performance Governor: By contrast there is the *performance* governor, that forces the CPU to always use the highest possible clock frequency. This frequency will be statically set and will not change. As such, this particular governor offers no power saving benefits. According to the “Mellanox performance tuning guide” [11], the performance of the InfiniBand cards can be improved by setting the governor to performance.

B. CPU Idle States

The x86-architecture CPUs support various states in which parts of the CPU are deactivated or run at lower performance settings. These states, known as *c-states* [12], are numbered from C0 upward, with higher numbers representing decreased CPU functionality and greater power saving. The c-states of a given number are broadly similar across processors, although the exact details of the specific feature sets of the state may vary between processor families. The more CPU units are stopped, the more energy is saved, and the more the time required for the CPU to wake up and be again 100% operational. Thus, enabling c-states can be a relevant latency source.

C. Nonuniform Memory Access

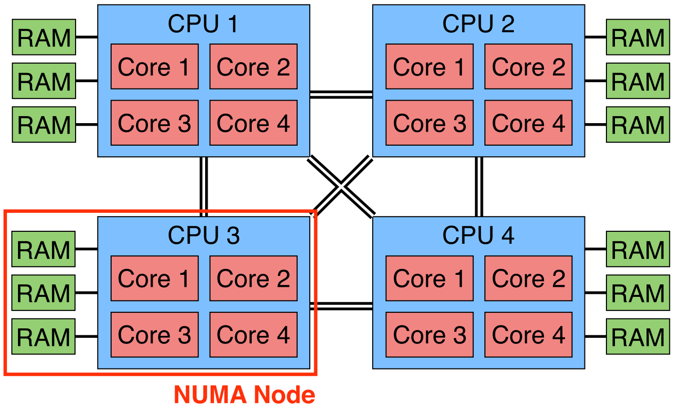

Fig. 4. Example of an NUMA architecture.

Modern servers with two or more processors commonly have a nonuniform memory access (NUMA) architecture, meaning that there are different memory performance and latency characteristics when accessing memory local to one processor, also called NUMA node, than when accessing memory directly attached to another processor in the same server (see Fig. 4). Similarly, PCIe I/O devices are local to a specific processor and remote with respect to the others. To access a remote NUMA node, the memory request must traverse the inter-CPU link and use the memory controller associated with the remote NUMA node. This incurs a latency penalty on a remote NUMA node memory access. In order to run an application on a certain NUMA node, the process affinity should be set using tools, such as the *numactl* utility [13], which can control the NUMA policy for processes or shared memory.

IV. Performance and Scalability Tests

We had the opportunity to perform some scalability tests with LSEB on Galileo [14], one of the main supercomputers hosted at CINECA [15]. CINECA is a nonprofit Consortium of participants including 70 Italian universities, four Italian research institutions and the Italian Ministry of Education. It offers support to the research activities of the scientific community through supercomputing and its applications. Galileo is the Tier-1 system of CINECA, introduced on January 2015. It is devoted to scientific computing on the basis of national and European proposals. This supercomputer is used to optimize and develop applications targeted at hybrid architectures, leveraging software applications in the fields of computational fluid dynamics, material and life science, and geophysics. The computing system is also available to the European researchers as a Tier-1 system of the PRACE infrastructure [16]. Galileo is composed of 516 compute nodes, each containing two Intel Xeon Haswell 8-core E5–2630 v3 processors, with a clock of 2.40 GHz. All the compute nodes have 128 GB of memory; 384 compute nodes are also equipped with two accelerators (Intel Xeon Phi 7120P), for a total of 768 accelerators in the whole system. Each node is connected to an InfiniBand network through a QLogic Single-Port quad data rate (QDR) InfiniBand HCA [17]. The datasheet of this HCA model specifies a bandwidth of just 27.2 Gb/s, instead of the 32 Gb/s foreseen by the standard. For this reason, we assume a maximum theoretical bandwidth of 27.2 Gb/s. Table I summarizes the system specification of Galileo.

TABLE I
GALILEO SYSTEM SPECIFICATION

Model	IBM NeXtScale
Architecture	Linux InfiniBand Cluster
Nodes	516
Processors	2 x 8-cores Intel Haswell 2.40 GHz
Cores	16 cores/node, 8256 cores in total
Accelerators	2 Intel Phi 7120p per node on 384 nodes
RAM	128 GB/node, 8 GB/core
Internal Network	InfiniBand with 4x QDR switches
HCA	QLogic Single-Port QDR InfiniBand

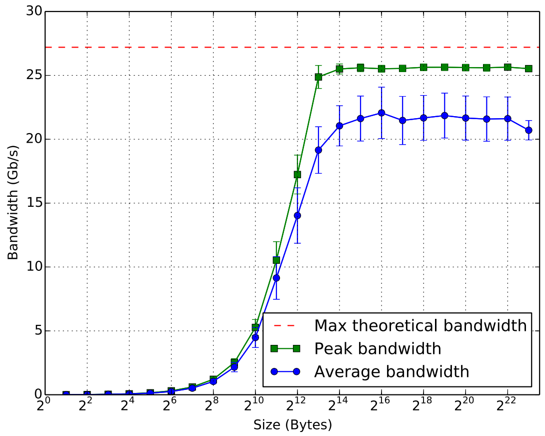

Fig. 5. Benchmark with ib_write_bw on Galileo: the blue line represents the resulting average bandwidth and the green line represents the peak bandwidth.

Due to the policies of the cluster, it was not possible to apply all the fine-tuning operations described in Section III in order to achieve the best performance. Indeed for reasons related to power consumption, in Galileo, each node has the governor set to "ondeman" and the c-states enabled. However, it was possible to take advantage of the process affinity using *numactl*.

A. Standard Benchmark

Generically speaking, before testing an application on one or more nodes, it is recommended to execute a standard benchmark in order to establish a performance baseline. In the case of applications that make use of RDMA, there is a set of microbenchmarks provided by the OpenFabrics Enterprise Distribution package [18] that allows us to verify the effective point-to-point network capacity. One of these microbenchmarks, the so-called ib_write_bw, was chosen and used to identify the real maximum bandwidth attainable between two random nodes belonging to the cluster. The tests performed with ib_write_bw foresee the execution of 5000 bidirectional RDMA write transactions for each different buffer size from 2 to 2^{22} B.

In Fig. 5, the average and the peak bandwidths obtained running ten times ib_write_bw on two Galileo nodes are plotted as a function of the buffer size. The peak bandwidth refers to the maximum bandwidth measured among the 5000 transactions foreseen by the benchmark. It is important to note that the average bandwidth is affected by a significant statistical error. Moreover, the average bandwidth and the peak bandwidth differ by about 15%. These behaviors are mostly caused by the constant presence of external jobs running on the cluster and by the missing optimization settings on the nodes. Considering the average bandwidth, the benchmark reaches 21.4 Gb/s, that is, 78.8% of the maximum theoretical bandwidth expected.

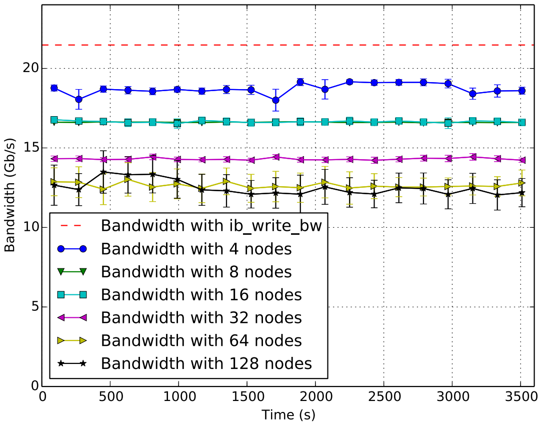

Fig. 6. Bandwidth measurements for LSEB on an increasing number of nodes on Galileo.

B. Results With Event Builder Software

As it is typical for clusters offering concurrent access to shared resources, Galileo computing nodes are accessible exclusively through a job scheduler. A job is described by providing a list of computational requirements on a limited set of resources, such as the number of needed cores to run on. Based on Galileo, the maximum limit for the usable number of cores is unfortunately set to 1024; thus, although more than 500 nodes are available on the system, not all of them can be used concurrently by the same application. Even asking for eight cores on each node instead of the 16 available, the maximum number of allocable nodes is 128. Reducing the number of required cores per node may increase the number of allocable nodes but also may increase the possibility to have external jobs running on the same nodes at the same time, which would disturb significantly our measurements. Starting from a four-node setup, several tests were performed, doubling the number of nodes at every test, up to 128. On each node, LSEB could run on just two cores, one for the RU thread and one for the BU thread. However, in order to avoid the concurrent execution of external processes, as discussed earlier, it is desirable to reserve all the available cores on each node. This was indeed done for the tests up to 32 nodes. For the tests with 64 and 128 nodes, instead, only half of the 16 cores available on each node were allocated. In all the tests, LSEB runs with the same fragment aggregation cardinality of 600 fragments and an average fragment size of 200 B. This implies that each data transfer has an average size of about 128 kB.

The average bandwidths measured running LSEB on Galileo in different node configurations are reported in Fig. 6 as a function of time, presenting good stability over time. The scalability plot is shown in Fig. 7, where the average bandwidth is plotted as a function of the number of nodes; the bandwidth

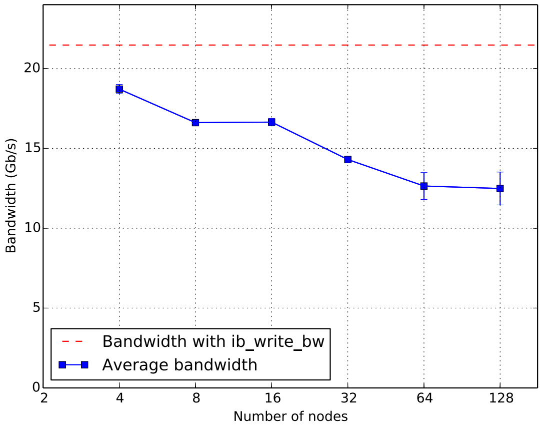

Fig. 7. Bandwidth scalability for LSEB on an increasing number of nodes on Galileo.

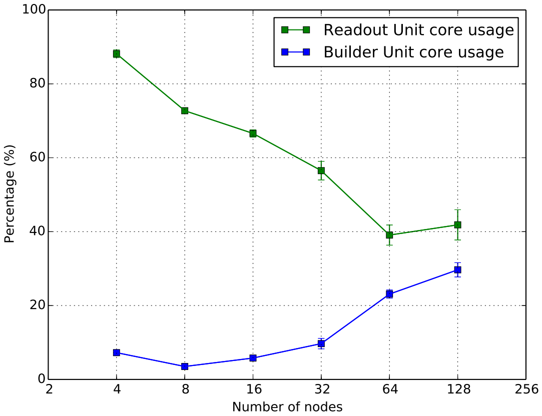

Fig. 8. RU and BU core usage for LSEB on an increasing number of nodes on Galileo.

slowly decreases as the number of nodes increases, reaching about 58% of the benchmarked bandwidth for 128 nodes. The average core usage of the RUs and the BUs is plotted as a function of the number of nodes in Fig. 8. The core usage of the RU shows a decreasing trend as the number of nodes increases, which is caused by the increment of time spent waiting for the completion of transfer operations. On the other hand, the core usage of the BU has an increasing trend, caused by the increment of the number of fragments that needs to be received and checked for each event. However, the BU does not saturate the core even with 128 nodes, reaching a load of about 30%.

V. Investigating Low Power Architectures

As described previously, LSEB is not a CPU consuming application, so we investigated also the possibility to implement a high-throughput LAN using low-power servers instead of the traditional ones. The technical design report for the trigger and online upgrade of LHCb [1] assumes that a single node of the EB farm can need up to 400 W, with 130 W required only by the PCIe40 board [3]. With the foreseen size of the EB farm, these power consumptions can lead to relevant electricity and cooling costs. In this section, results of a preliminary study are presented; the scalability has not been studied but just the performance and the power consumption with few nodes. For this reason, the testbed is composed of two nodes of the same type connected back-to-back with InfiniBand HCAs.

Measurements on low-power architectures are performed within the *Computing On System on Chip (SoC) Architectures* Project [19] collaboration, that provided the testbed. The tested architectures are the Intel Xeon D-1540 [20] and the Intel Atom C2750 [21], two low-power x86 processors designed by Intel. The available InfiniBand HCAs were the QLogic QDR InfiniBand HCA and the Mellanox fourteen data rate (FDR) InfiniBand HCA [22]. The QLogic QDR InfiniBand HCA is the same model of HCA tested in Section IV, it requires a PCIe-2 slot with eight lanes and provides a maximum bandwidth of 27.2 Gb/s. The Mellanox FDR InfiniBand HCA requires a PCIe-3 slot with 16 lanes and provides a bandwidth of 54.3 Gb/s. Depending on the PCIe capacity of each architecture, different InfiniBand HCAs were used. The results are compared with those obtained on a testbed composed of two Intel Xeon E5-2683v3 (14 cores, 2 threads per core, 35-MB cache, base frequency 2 GHz, and turbofrequency 3 GHz)-based standard servers [23].

The bandwidths presented in this section are average bandwidths obtained running each measurement ten times. Tests performed are similar to those described in Section IV, with the addition of the power consumption monitoring while running LSEB. These measurements refer to the overall power consumption of each node, including processor, motherboard, memory, disk, and InfiniBand network card. The c-states feature (see Section III-B) was active in all tests performed, because disabling it would result in a higher power consumption.

In all the tests, LSEB runs with the same fragment aggregation cardinality of 600 fragments and an average fragment size of 200 B. This implies that each data transfer has an average size of about 128 kB.

A. Intel Xeon D-1540

The Intel Xeon D-1540 (XeonD from now on) brings the performance of Intel Xeon processors into a lower power SoC. The XeonD was released by Intel in the first quarter of 2015. It is a low-power SoC designed for servers and manufactured on 14-nm technology. The XeonD has eight cores (with two threads per core) and it supports PCIe-3 (with max 24 lanes) and PCIe-2 (with max eight lanes) interconnects. Being more a server than a pure low-power architecture, the Intel Xeon D-1540 has a higher power consumption with respect to the Intel Atom C2750, with a thermal design power (TDP) of 45 W. This processor is mounted on a Super Micro X10SDV-F motherboard [24] that provides a PCIe-3 slot with 16 lanes. In this testbed, the two XeonD nodes are connected together with Mellanox FDR InfiniBand cards.

1) Standard Benchmark: The benchmark performed shows that both the Xeon servers and the XeonD nodes can reach the maximum bandwidth available when using the Mellanox FDR cards, as shown in Figs. 9 and 10, respectively. The Xeon servers achieve 99.57% of the theoretical bandwidth and the XeonD nodes reach 99.35%. Furthermore, changing the governor settings does not affect the performance.

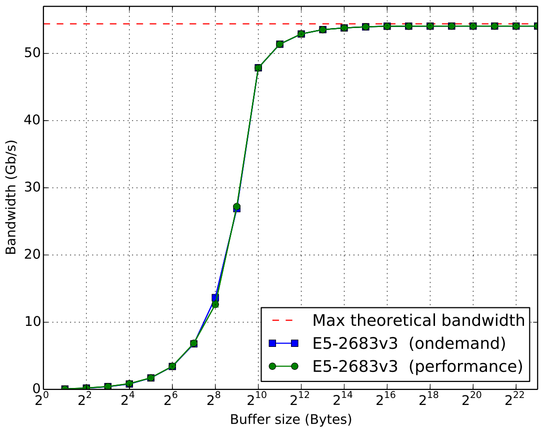

Fig. 9. Benchmark with ib_write_bw on E5-2683v3 nodes with FDR cards.

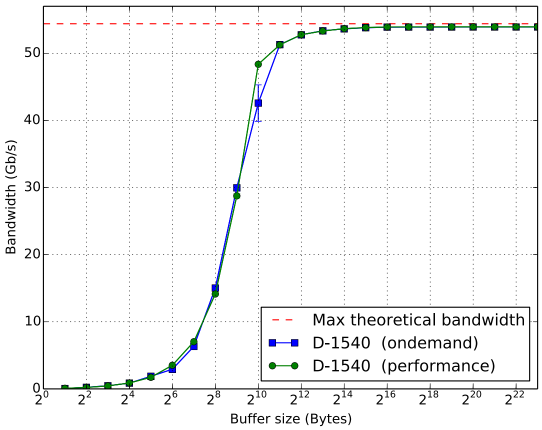

Fig. 10. Benchmark with ib_write_bw on D-1540 nodes with FDR cards.

2) Results With Event Builder Software: The measurements of the bandwidth running LSEB include the same tests performed with the Atom and an additional test which adds a pure computation process running over four cores on each node during the second half of the run. The aim of this added test is to investigate the possibility of the EB to perform computations, such as a preanalysis of the events prior to send them to the EFF.

The bandwidths as a function of time are shown in Fig. 11, where the black vertical line indicates the computation process start time. Both the Xeon servers and the XeonD nodes reach about 99.12% of the theoretical bandwidth allowed by the FDR InfiniBand cards, keeping the bandwidth stable even with a computation process running. These performances were achieved with very different power consumptions, with the XeonD nodes that require about a third of the energy consumed by the Xeon servers, as shown in Fig. 12. Table II summarizes all the measurements performed in this testbed. The power consumption in idle state and the maximum temperature of each type of processor are also reported in Table II.

B. Intel Atom C2750

The Intel Atom C2750 was released by Intel in the third quarter of 2013. It is a low-power SoC designed for microservers and manufactured on 22-nm technology.

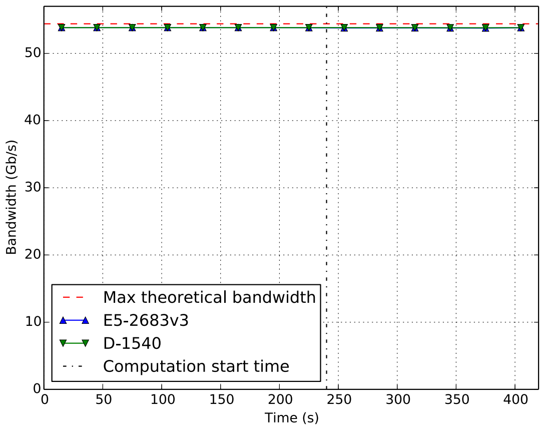

Fig. 11. Bandwidth running LSEB on nodes with FDR cards.

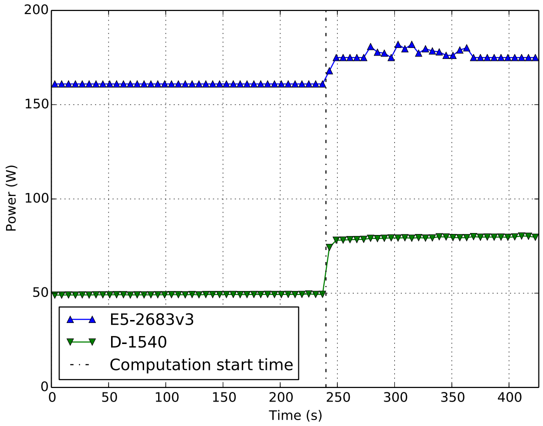

Fig. 12. Power consumption running LSEB on nodes with FDR cards.

TABLE II
POWER CONSUMPTION, TEMPERATURE, AND BANDWIDTH MEASURED RUNNING LSEB ON NODES WITH FDR CARDS

	E5-2683v3	**D-1540**
Idle power consumption	80.78 W	28.23 W
LSEB power consumption	161.00 W	49.02 W
LSEB power consumption with computation	176.54 W	79.12 W
Max temperature	56.0 C	59.0 C
Average bandwidth	53.82 Gb/s	53.82 Gb/s

The Intel Atom C2750 has eight cores without multithreading and it supports PCIe-2 interconnects with a maximum of 16 lanes. The TDP of the Intel Atom C2750 is of 20 W. This processor is mounted on a Super Micro A1SAi-2750F motherboard [25] that provides a slot PCIe-2 with eight lanes. In this testbed, the two Atom nodes are connected together with QLogic QDR InfiniBand cards, due to the missing PCIe-3 support required by the more powerful Mellanox FDR InfiniBand cards.

1) Standard Benchmark: The results obtained running ib_write_bw on the Xeon servers with QDR InfiniBand connectivity are shown in Fig. 13, where the bandwidth is plotted as a function of the buffer size for both the governor settings.

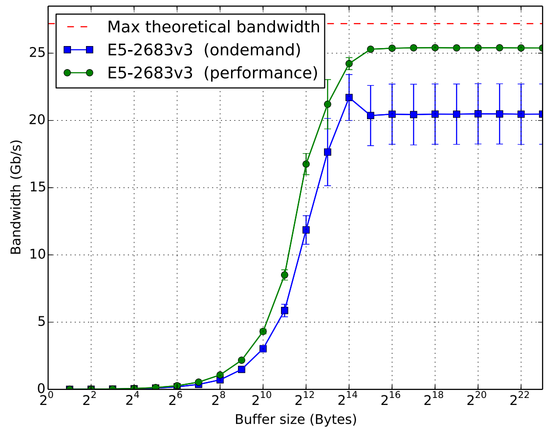

Fig. 13. Benchmark with ib_write_bw on E 5-2683v3 nodes with QDR cards.

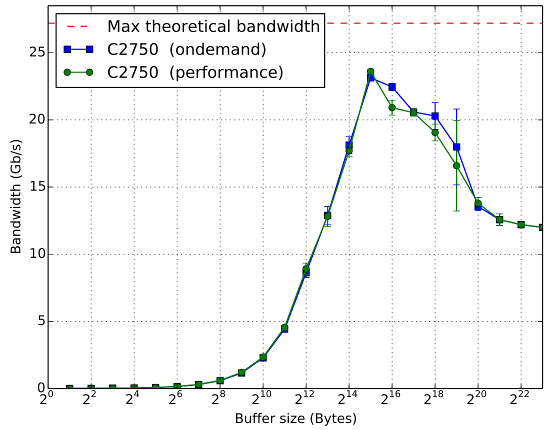

Fig. 14. Benchmark with ib_write_bw on C 2750 nodes with QDR cards.

This benchmark confirms the behavior observed testing LSEB on the Galileo cluster reported in Section IV: setting the governor to "ondemand" does not permit reaching the best performance and a stable bandwidth. On the other hand, a higher and more stable bandwidth is obtained setting the governor to "performance." In this configuration, the maximum average bandwidth reached is 25.41 Gb/s, that is, 93.42% of the maximum theoretical bandwidth foreseen by this kind of interconnects. We presume that this difference of performance using different governors is mostly related to the specific InfiniBand HCA used rather than the CPU technology. In fact, this behavior is not present using the Mellanox FDR InfiniBand HCA, as seen in Section V-A1.

On the Atom nodes, the measured bandwidths present a relevant degradation with buffer sizes greater than 32 kB using both the governor settings, as shown in Fig. 14. This behavior can be caused by several factors, such as CPU inefficiencies or network card issues with this platform. A further test with different network cards may help to fully understand this degradation in performance.

2) Results With Event Builder Software: Apart from the benchmarks, the performance inefficiency of the Atom nodes is unveiled also running LSEB. Fig. 15 shows that the Atom nodes reach an average bandwidth of 15.37 Gb/s with respect to the 23.19 Gb/s reached by the Xeon servers.

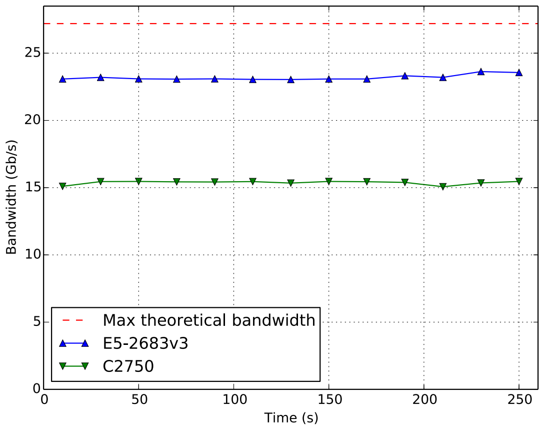

Fig. 15. Bandwidth running LSEB on nodes with QDR cards.

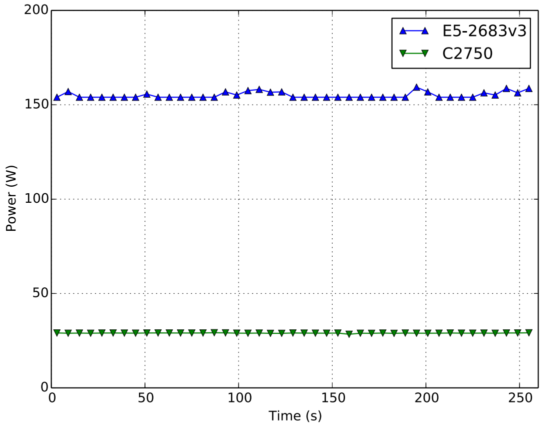

Fig. 16. Power consumption running LSEB on nodes with QDR cards.

TABLE III

POWER CONSUMPTION, TEMPERATURE, AND BANDWIDTH MEASURED RUNNING LSEB ON NODES WITH QDR CARDS

	E5-2683v3	**C2750**
Idle power consumption	77.46 W	18.20 W
LSEB power consumption	154.44 W	28.93 W
Max temperature	52.0 C	37.0 C
Average bandwidth	23.19 Gb/s	15.37 Gb/s

On the other hand, the power consumption measurements turn the tables: the Atom nodes consume about 18.73% with respect to the Xeon servers (28.93 W against 154.44 W), as shown in Fig. 16. A rough metric that could be used to compare these results is the energy consumption rate (ECR) expressed in W/Gbps, that is the power consumption in Watt spent for each Gb/s of bandwidth. Following this metric, the Atom nodes are three times more performant with respect to the Xeon servers, showing an ECR of 1.88 and 6.66 W/Gbps, respectively. With regard to the previous testbed, the ECR values of the XeonD and the Xeon servers using the Mellanox FDR InfiniBand HCA are 0.93 and 2.99 W/Gbps, respectively.

Table III summarizes all the measurements performed in this testbed. The power consumption in idle state and the maximum temperature of each type of processor are also reported in Table III.

VI. Conclusion

LSEB was successfully tested on up to 128 nodes of the Galileo cluster that is composed of more than 500 nodes interconnected with QDR InfiniBand cards. For cost reasons, the cluster is tuned to save energy rather than to achieve the best performance. Nevertheless, the presence of hundreds of nodes allowed us to perform tests at a scale similar to that foreseen by the LHCb upgrade. Tests have shown that LSEB scales up to 128 nodes reaching 60% of the available point-to-point bandwidth. Moreover, the RU and the BU use little CPU capacity.

A secondary activity has been the evaluation of x86-based low-power boards, to assess their suitability for the event-building process foreseen by LHCb. Two different boards were tested: the Intel Atom C2750 and the Intel Xeon D-1540. The measured bandwidth and power consumption were compared with those obtained on standard servers. The tests have shown that the Intel Xeon D-1540 achieves results that are comparable in terms of bandwidth with those of a standard server but with a third of the power consumption. On the other hand, the Intel Atom C2750 is not able to achieve the high performance provided by standard Intel Xeon processors; however, its extremely low power consumption makes it still interesting for data acquisition purposes where the bandwidth and computational requirements are less strict than those of the event-building of the LHCb experiment.

References

[1] LHCb Collaboration, "LHCb trigger and online upgrade technical design report," CERN, LHCC, Prévessin-Moëns, France, Tech. Rep. C10026, 2014.

[2] P. Durante *et al.*, "100 Gbps PCI-express readout for the LHCb upgrade," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 4, pp. 1752–1757, Aug. 2015, doi: 10.1109/TNS.2015.2441633.

[3] J.P. Cachemiche *et al.*, *The PCIe-Based Readout System for the LHCb Experiment*, document JINST 11 P02013, 2016, doi: 10.1088/1748-0221/11/02/P02013.

[4] A. Otto *et al.*, "A first look at 100 Gbps LAN technologies, with an emphasis on future DAQ applications," *J. Phys. Conf. Ser.*, vol. 664, no. 5, p. 052030, 2015, doi: 10.1088/1742-6596/664/5/052030.

[5] R. Buyya, T. Cortes, H. Jin, *An Introduction to the InfiniBand Architecture*. Hoboken, NJ, USA: Wiley, 2002.

[6] *100 Gigabit Ethernet Technology Overview*, accessed on Apr. 10, 2017. [Online]. Available: http://www.ethernetalliance.org/wp-content/uploads/2011/10/document_files_40G_100G_Tech_overview.pdf

[7] *Intel Omni-Path Architecture: The Next-Generation Fabric*, accessed on Apr. 10, 2017. [Online]. Available: http://www.intel.com/content/www/us/en/high-performance-computing-fabrics/omni-path-architecture-fabric-overview.html

[8] M. Manzali, *Lseb 2.0*, doi: 10.5281/zenodo.46935.

[9] Hilland. (2003). *RDMA Protocol Verbs Specification, Version 1.0*, accessed on Apr. 10, 2017. [Online]. Available: http://www.rdmaconsortium.org/home/draft-hilland-iwarp-verbs-v1.0-RDMAC.pdf

[10] *Red Hat Enterprise Linux 6—Power Management Guide*, accessed on Apr. 10, 2017. [Online]. Available: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Power_Management_Guide/index.html

[11] (2016). *Performance Tuning Guide for Mellanox Network Adapters, Revision 1.17*, accessed on Apr. 10, 2017. [Online]. Available: https://www.mellanox.com/related-docs/prod_software/Performance_Tuning_Guide_for_Mellanox_Network_Adapters_Archive.pdf.

[12] J. Kukunas, "Intel core processors," in *Power and Performance: Software Analysis and Optimization*. San Mateo, CA, USA: Morgan Kaufmann, 2015, pp. 43–52.

[13] Andi Kleen. (2004). *An NUMA API for Linux*. [Online]. Available: http://halobates.de/numaapi3.pdf

[14] *GALILEO Cluster*, accessed on Apr. 10, 2017. [Online]. Available: http://www.hpc.cineca.it/hardware/galileo

[15] *CINECA Consortium*, accessed on Apr. 10, 2017. [Online]. Available: http://www.cineca.it/en

[16] *PRACE Research Infrastructure*, accessed on Apr. 10, 2017. [Online]. Available: http://www.prace-ri.eu/

[17] *QLogic QLE7340 Datasheet*, accessed on Apr. 10, 2017. [Online]. Available: http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/truescale-infiniband-qle7300-brief.pdf.

[18] *OpenFabrics Alliance*, accessed on Apr. 10, 2017. [Online]. Available: https://www.openfabrics.org/index.php/openfabrics-software.html

[19] *INFN COSA (Computing On Soc Architectures) Project*, accessed on Apr. 10, 2017. [Online]. Available: http://www.cosa-project.it/home.html

[20] *Intel Xeon Processor D-1540*, accessed on Apr. 10, 2017. [Online]. Available: http://ark.intel.com/products/87039/Intel-Xeon-Processor-D-1540-12M-Cache-2 00-GHz

[21] *Intel Atom Processor C2750*, accessed on Apr. 10, 2017. [Online]. Available: http://ark.intel.com/products/77987/Intel-Atom-Processor-C2750-4M-Cache-2 40-GHz

[22] *Mellanox Connect-IB Single-Port InfiniBand HCA Product Brief*, accessed on Apr. 10, 2017. [Online]. Available: http://www.mellanox.com/related-docs/prod adaptercards/PB_Connect-IB.pdf

[23] *Intel Xeon Processor E5-2683 v3*, accessed on Apr. 10, 2017. [Online]. Available: http://ark.intel.com/products/81055/Intel-Xeon-Processor-E5-2683-v3-35M-Cache-200-GHz

[24] *Super Micro X10SDV-F Motherboard*, accessed on Apr. 10, 2017. [Online]. Available: http://www.supermicro.com/products/motherboard/Xeon/D/X10SDV-F.cfm

[25] *Super Micro A1SAi-2750F Motherboard*, accessed on Apr. 10, 2017. [Online]. Available: http://www.supermicro.com/products/motherboard/atom/X10/A1SAi-2750F.cfm