EPJ Web of Conferences **245**, 01012 (2020) https://doi.org/10.1051/epjconf/202024501012
CHEP 2019

Network simulation of a 40 MHz event building system for the LHCb experiment

Flavio Pisani[1,2,3,*], *Tommaso* Colombo[3], *Niko* Neufeld[3], *Umberto* Marconi[2], *Rafal* Krawczyk[3], *Domenico* Galli[1,2], *Rainer* Schwemmer[3], and *Paolo* Durante[3]

[1]Alma Mater Studiorum – Università di Bologna, Bologna, Italy
[2]Istituto Nazionale di Fisica Nucleare – INFN, Sez. di Bologna, Bologna, Italy
[3]CERN, Geneva, Switzerland

Abstract. The LHCb experiment will be upgraded in 2021, and a new triggerless readout system will be implemented. In the upgraded system, both Event Building (EB) and event selection will be performed in software for every collision produced in every bunch-crossing of the LHC. In order to transport the full data rate of 32 Tb/s, we will use state of the art off-the-shelf network technologies, e.g. InfiniBand EDR. The full event building system will require around 500 nodes interconnected together via a non-blocking topology. Because of the size of the system it is challenging to test at production scale, before the actual procurement. Therefore, we resort to network simulations as a powerful tool for finding the optimal configuration. We developed an accurate low-level description of an InfiniBand based network with event building like traffic. We will present a full-scale simulation of a possible implementation of the LHCb EB network.

1 Introduction

The Large Hadron Collider beauty (LHCb) experiment [1] is receiving a substantial upgrade [2] during a planned maintenance shutdown of the Large Hadron Collider (LHC) called Long Shutdown 2 (LS2). One of the major changes during this upgrade process is the installation of a completely new Data Acquisition (DAQ) system without any low-level hardware trigger, allowing all event selection being performed by software-based algorithms providing high yields while keeping the contamination of unwanted events as low as possible.

To implement a triggerless readout, data must be forwarded by the event building network at the full rate produced by the detector. In the new upgraded configuration the aggregate data rate will be ∼32 Tb/s, therefore to achieve this total throughput, we are targeting a system composed of ∼500 custom DAQ cards interconnected together using 100 Gb/s networking technology. To take advantage of Commercial Off-The-Shelf (COTS) hardware, the DAQ boards are PCIe-based add-on cards and they will be installed into standard servers. In order to provide the required networking capabilities the event builder PCs will be equipped with high-speed network cards. A full architectural schema of the readout system of the LHCb experiment is shown in Figure 1.

*e-mail: flavio.pisani@cern.ch

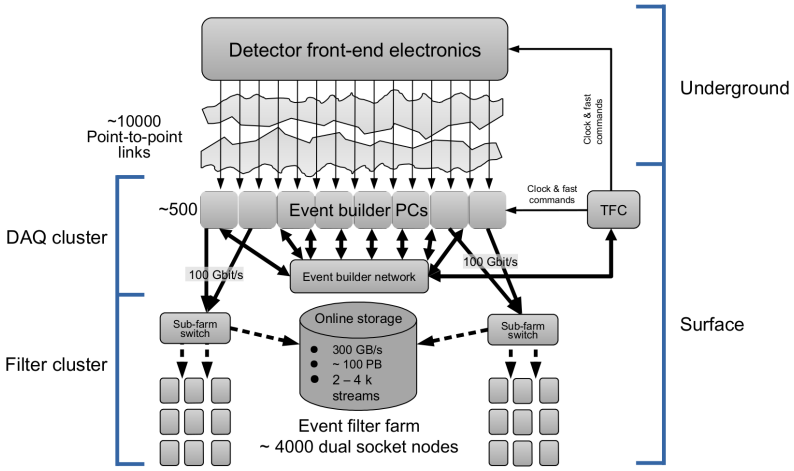

Figure 1. The architecture of the upgraded LHCb readout system. This diagram depicts the data flow from the front-end electronics, located 100 m underground, up to the long term data storage. This paper will investigate the architecture of the event builder, a more comprehensive description of the full DAQ chain can be found in [2].

After an extensive evaluation of multiple commercially available network technologies we selected InfiniBand to be our DAQ network candidate. InfiniBand is a network protocol standardised by the InfiniBand Trade Association (IBTA), and widely adopted by the High Performance Computing (HPC) community. Over the years the theoretical link throughput of this technology has been increased from the 10 Gb/s provided by InfiniBand SDR to the 200 Gb/s provided by InfiniBand HDR. Given the throughput requirements of the experiment, at the time of writing, we are targeting InfiniBand EDR – i.e. the 100 Gb/s variant of InfiniBand.

The process of designing and building a system with the above-mentioned complexity requires extensive planning and testing. For this reason we developed two event building benchmarks: DAQ Protocol-Independent Performance Evaluator (DAQPIPE) and All-to-All (a2a). Those event-building benchmark applications generate real event building traffic and they can be configured in multiple ways, in order to experiment with different network configurations and technologies on existing infrastructures.

In order to overcome the lack of suitable clusters for testing, we developed an accurate low-level simulation model that replicates an InfiniBand-based interconnection network. This simulation model can be used to detect criticalities and issues that may arise on real systems, resulting in more a optimised use of the limited time available at existing HPC facilities. Moreover, it is possible to simulate non-existing systems providing a valuable input in the design process of the real system. A full description of the simulation model and its validation against data gathered on real HPC clusters can be found in [3].

In this work, we present the impact of different interconnection networks on the a2a benchmark, and a full-scale test of the DAQPIPE application.

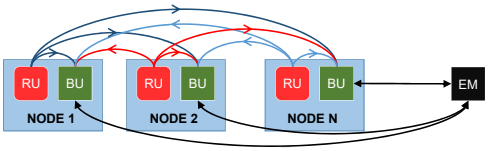

Figure 2. The logic representation of the LHCb Event Building process. The different arrows represent the multiple fragments sent by the Readout Unit and gathered by the Builder Unit while the black ones the control messages to and from the Event Manager.

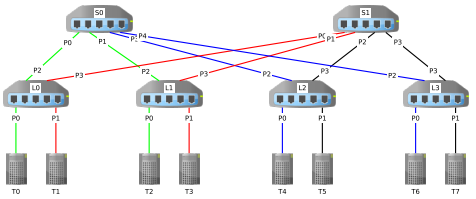

Figure 3. Example of a fat-tree network architecture built using switches with a radix of four. The two switches in the upper part are called spine switches, while the four in the lower part are called leaf switches. The different colours represent the paths selected by the optimised routing algorithm for different packets.

2 The LHCb Event Building architecture

2.1 The Event Building process

As described in [2] the data flow of the LHCb detector is divided across ~ 10000 optical fiber links terminated onto ~ 500 custom FPGA-based readout readout cards.

In order to collect data from all the different sub-detector channels, all the data fragments of a single event have to be collected and assembled in the same place, resulting in the EB process. Therefore all the different event fragments must be sent over some interconnection network in an all-to-one communication. Three logical units can be defined to provide a high-level description of this task: Builder Units, Readout Units and the Event Manager; a functional description follows:

- **Readout Unit (RU)** collects the fragments from the PCIe-based DAQ board and sends them to the Builder Units (BUs)
- **Builder Unit (BU)** receives and aggregates the fragments into full events
- **Event Manager (EM)** assigns which event is built on which BU

All the EB nodes (i.e. the servers that are hosting the required hardware and running the EB software) have two units: one RU and one BU, as depicted in Figure 2; because data always flows from the RUs to the BUs, a "folded" structure can profit from the full-duplex nature of COTS network technologies; and it leads to a reduction by a factor two in the number of physical machines used.

The HPC community identifies recurrent complex communication patterns as collective communications [4], in this context the traffic pattern of a folded event builder can be mapped to an All-To-All Personalised Communication (ATAPC) with different data size for every fragment.

2.2 Traffic shaping and network topology

In order to reduce network congestion generated by an ATAPC, we took advantage of the extensive work performed by the HPC community like the one in [5] or on the very similar All-To-All Broadcast in [6], and we decided to use the *linear shifting* traffic scheduling technique. This traffic shaping technique can be explained as follows:

- We divide the ATAPC into N phases, where N is the total number of nodes
- In every phase, every node sends data to exactly one destination and receives data from one source only
- During phase n node i sends to node $(n + i)\%N$[1]
- When a previously agreed condition is met all the nodes synchronously switch from phase n to phase $n + 1$, usually the switching is triggered either by the number of sent events or by a fixed time window

If the aforementioned conditions are respected for all the phases, then we have an ideal linear shifting scheduling. Fulfilling all the conditions requires strong time synchronisation. However, to provide better scalability on large systems, it is desirable to tolerate a temporary phase desynchronisation. The performance degradation introduced by a violation of the scheduling policy can be absorbed into the operational margin of the individual links, in particular the average and maximum throughput required by every RU/BU are 64 Gb/s and 80 Gb/s respectively; therefore a temporary phase desynchronisation of less than 20% can be tolerated. In any case the scheduler should prevent a persistent phase shifting.

The use of the aforementioned scheduling technique provides an easy way to solve the destination conflict issue generated by the event building traffic. On the other hand, it requires the processing of multiple events in parallel; some degree of synchronisation among the nodes; and it does not guarantee end-to-end congestion-free network traffic. In order to have the certainty of no resource contention, the underlying network has to be *rearrangeably non-blocking*.

A folded Clos network [7] such as the one depicted in Figure 3 fulfils the aforementioned requirements, and therefore we decided to use this topology for the LHCb DAQ network. This topology, often referred to as fat-tree[2], it is widely adopted, and switch vendors support it; in particular, the OpenSM subnet manager used in InfiniBand-based networks provides a routing algorithm [8] optimised for linear shifting traffic.

2.3 The Event Building benchmarks

In order to design and optimise the EB system before the commissioning of the detector, we developed different traffic generators that replicate the expected EB traffic. Those applications generate real network traffic on existing computing clusters, and are used to validate various aspects of the EB design. We developed two main benchmarks that try to address the problem of building events in different ways. It is important to note that those benchmarks could be used as the communication core for the actual EB application, if fully integrated with the experiment's software ecosystem.

DAQPIPE

DAQPIPE [9][10][11] is a small benchmark application to test network fabrics. It emulates an event builder based on a local area network, and it supports multiple network technologies through different communication libraries. The software provides an implementation of all the logical blocks described in Sect. 2 and emulates reading data from a real DAQ board connected to the detector.

[1] The % symbol indicates the modulo operation

[2] From a rigorous point of view, the network topology shown in Figure 3 is a folded Clos network, nevertheless, in the industry and data centre world, it is frequently referred to as fat-tree. Even if the network topologies are not the same from this point on we will use the industry-standard naming 'fat-tree' instead of 'folded Clos'.

This benchmark uses a linear shift-like traffic shaping, without enforcing strong synchronisation among the nodes[3]. The lack of fully synchronous scheduling creates temporary saturation on specific links. This effect is mitigated by sending multiple fragments to different destinations at the same time from every RU, the number of fragments in flight per RU and the number of events processed in parallel can be configured via two parameters:

- **Credits:** number of events processed in parallel by the BU
- **Parallel sends:** number of fragments of the same event in flight to the same BU from a different RUs

The *credits* are processed in a fully asynchronous way, and they mitigate the impact of temporary link congestion and latency toward the EM. The *parallel sends* define a sliding window in the scheduling absorbing the latency between BUs and RUs.

a2a

a2a is a micro-benchmark recently developed to test how a fully synchronous linear shifter approach will perform in an event building scenario; therefore a2a faces the problem from an opposite point of view compared to DAQPIPE. At the time of writing this benchmark enforces strict phase synchronisation via a time-based method, this strategy has been initially preferred to in-band signalling to reduce the communication latency penalty.

The traffic scheduling can be configure in a2a via two parameters:

- **Transmission window:** amount of time in which every RU is allowed to send fragments to the target BU selected according to the linear shifting scheduling
- **Idle window:** period of time in which no RUs is supposed to send

The *Transmission window* selects the maximum amount of data exchangeable in one phase, while the *idle window* absorbs any desynchronisation that may occur between the various nodes. The ratio of the two windows defines the target throughput fraction T_f of the application:

$$T_f = \frac{T_w}{I_w + T_w} \tag{1}$$

where T_w and I_w represents the *transmission window* and the *idle window* respectively.

3 Results

In this section, we present the results of low-level network simulations of two different scenarios relevant to the LHCb event building. Those simulations are performed using the simulation model described in [3] implemented using the Objective Modular Network Testbed in C++ (OMNeT++) framework [12].

3.1 a2a congestion simulation

The main disadvantage of a linear shifting scheduling technique is a strong dependency from the network infrastructure's capability of providing a contention-free operation. The final system will be designed to operate in optimal conditions; nevertheless, it is essential to evaluate the impact potential partial failures.

Figure 4 shows how the a2a benchmark is impacted by a network infrastructure that does not allow a conflict-free linear shifting scheduling; the data presented in the plot refer to

[3]There is a version of DAQPIPE with enforced timing, but it will not be considered for the purpose of this paper

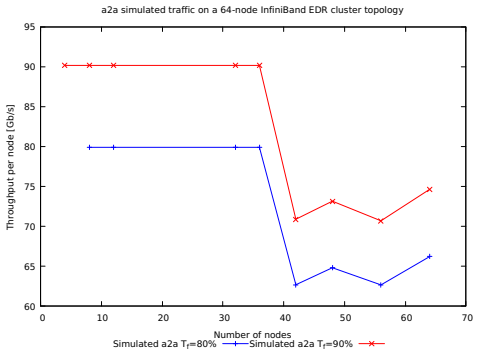

Figure 4. Simulation of the a2a benchmark on a simulated replica of a real HPC 64 nodes cluster. The plot depicts the minimum node throughput vs the number of nodes for target throughput fraction: $T_f = 80\%$ in blue and $T_f = 90\%$ in red.

the network simulation of a real HPC cluster network infrastructure which is not optimised for EB applications. When the number of nodes increases the probability of having resource contention in the network rises; the throughput drops significantly in presence of link oversaturation, because of the way the application is designed changing T_f does not have a positive effect on performance. This last effect is expected, and it is due to the strict implementation of the I_w, which does not allow any RU to send data. On the other hand, the application shows perfect scalability when there is no resource contention on the network; this behaviour has been verified on real systems multiple times.

3.2 Full scale DAQPIPE simulation

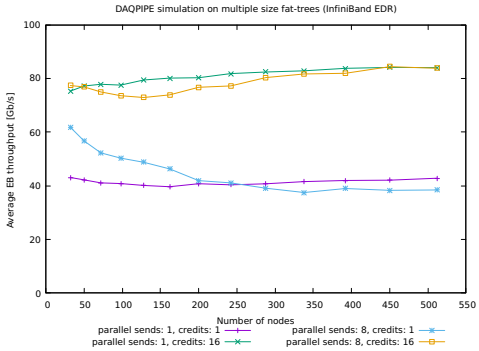

Figure 5. Average EB throughput of simulated DAQPIPE vs the number of nodes. All the network topologies tested are full fat-trees; the different node count is achieved by changing the switch radix.

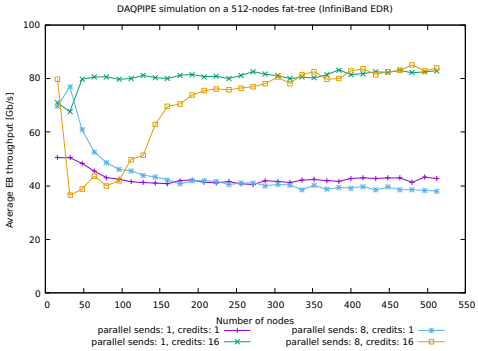

Figure 6. Average EB throughput of simulated DAQPIPE vs the number of nodes. The network topology tested is a 512-node fat-tree, the different node count is achieved by changing the number of leaf switches with active nodes connected.

Full-scale simulations of the traffic generated by DAQPIPE are fundamental to understand the behaviour of the application thoroughly. An *a priori* prediction of the application behaviour is nearly impossible for the following reasons: the performance variation introduced by a change in the number of credits or parallel sends are highly unpredictable; the application can generate non-trivial congestion patterns that need to be evaluated in simulation.

The scalability of the simulated DAQPIPE has been tested in two different scenarios: the first one was a test run on different fat-tree topologies with different number of nodes; the second one was simulation of a 512-node fat-tree with different number of active nodes. In order to keep the two configurations tested suitable for a linear shifting scheduling, the number of active nodes per leaf switch was kept homogeneous across the network topology.

The graph in Figure 5 presents the results obtained simulating the first of the two aforementioned scenarios. The simulations were executed with different parameter configurations in order to test the behaviour of DAQPIPE under different conditions. The configurations with 16 *credits* show solid performance in term of scaling and throughput, with the best configuration delivering more than 80 Gb/s in most of the tests. On the other hand, the configurations with only one credit suffer from the lower number of fragments being transferred in parallel, and the higher latency in the communication path with the EM; in particular the configuration with 1 *credit* and 8 *parallel sends* delivers high throughput in small systems, and experiences performance degradation in larger ones.

The plot in Figure 6 shows the results obtained simulating the second of the two aforementioned scenarios. The results under those different testing conditions show a more irregular behaviour, with a high throughput variability, especially in the 16 *credits* 8 *parallel sends* configuration. This behaviour can be explained by the fact that the number of parallel sends generates inter-RU resource contention. It is important to note that the particular performance drop, i.e. the one that affects the 8 *parallel sends* 16 *credits* configuration, it is not easily predictable by static traffic analysis. On the other hand, the configuration with 16 credits and 1 parallel send show a consistent and above 80 Gb/s throughput, and it experiences only a small performance penalty when running with a meager number of nodes.

The simulations show that the scalability of DAQPIPE is solid and that the application can deliver more than 80 Gb/s at the scale required by the LHCb DAQ system.

4 Conclusions

We introduced the changes that will be made to DAQ system of the LHCb experiment in the current upgrade process. We discussed the two main strategies that we developed over the years to solve the problem of building events with a 40 MHz event rate at the LHCb experiment. Thanks to low level network simulations we verified that: an a2a-like event builder can fulfil the requirements if the underlying network infrastructure is optimised for a linear shifting traffic; a DAQPIPE-like approach can deliver the required throughput of 80 Gb/s per EB node with an appropriate number of *credits* and *parallel sends*.

In conclusion an InfiniBand-based 40 MHz event building system for the LHCb experiment can be implemented by using either an a2a-like or a DAQPIPE-like approach.

References

[1] The LHCb Collaboration et al, Journal of Instrumentation **3**, S08005 (2008)

[2] Tech. Rep. CERN-LHCC-2014-016. LHCB-TDR-016 (2014), `https://cds.cern.ch/record/1701361`

[3] T. Colombo, P. Durante, D. Galli, M. Manzali, U. Marconi, N. Neufeld, F. Pisani, R. Schwemmer, S. Valat, IEEE Transactions on Nuclear Science **66**, 1159 (2019)

[4] J. Duato, S. Yalamanchili, N. Lionel, *Interconnection Networks: An Engineering Approach* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002), ISBN 1558608524

[5] Y. Yang, J. Wang, IEEE Transactions on Parallel and Distributed Systems **11**, 261 (2000)

[6] A. Faraj, X. Yuan, P. Patarasuk, IEEE Transactions on Parallel and Distributed Systems **18**, 264 (2007)

[7] C. Clos, Bell Syst. Tech. J. **32**, 406 (1953)

[8] Z. Eitan, J. Gregory, K.D. J., L. Michael, Concurrency and Computation: Practice and Experience **22**, 217 (2010), https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.1527

[9] A. Otto, D.H.C. Pérez, N. Neufeld, R. Schwemmer, F. Pisani, Journal of Physics: Conference Series **664**, 052030 (2015)

[10] D.H.C. Pérez, R. Schwemmer, N. Neufeld, IEEE Transactions on Nuclear Science **62**, 1110 (2015)

[11] F. Pisani, D.H.C. Pérez, N. Neufeld, Journal of Physics: Conference Series **608**, 012029 (2015)

[12] A. Varga, R. Hornig, *An Overview of the OMNeT++ Simulation Environment*, in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops* (ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2008), Simutools '08, pp. 60:1–60:10, ISBN 978-963-9799-20-2, http://dl.acm.org/citation.cfm?id=1416222.1416290